**PCT** WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 13/00 — A2

(11) International Publication Number: **WO 00/27131**

(43) International Publication Date: 11 May 2000 (11.05.00)

(21) International Application Number: PCT/GB99/03584

(22) International Filing Date: 29 October 1999 (29.10.99)

(30) Priority Data:
9823689.6 30 October 1998 (30.10.98) GB

(71) Applicant *(for all designated States except US)*: THE UNIVERSITY COURT OF THE UNIVERSITY OF GLASGOW [GB/GB]; University Avenue, Glasgow G12 8QQ (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: JIN, Joseph, Zhengping [GB/GB]; 75 Woodvale Avenue, Glasgow G61 2NX (GB). NIBLETT, Timothy, Bryan [GB/GB]; 4 Turnberry Road, Glasgow G11 5AE (GB). URQUHART, Colin, William [GB/GB]; Flat T/R, 7 Niddrie Square, Glasgow G42 8QX (GB).

(74) Agents: MCCALLUM, William, Potter et al.; Cruikshank & Fairweather, 19 Royal Exchange Square, Glasgow G1 3AE (GB).

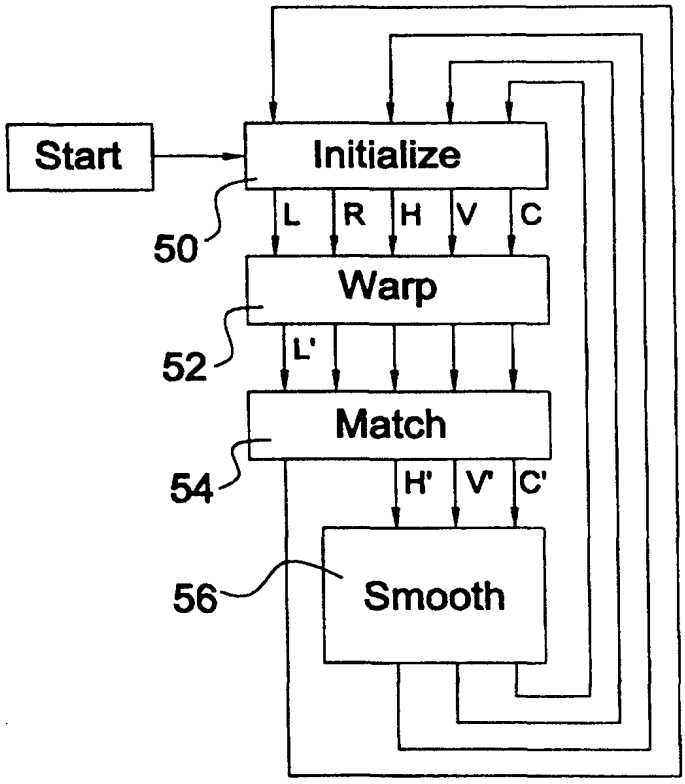
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published**
*Without international search report and to be republished upon receipt of that report.*

(54) Title: IMPROVED METHODS AND APPARATUS FOR 3–D IMAGING

(57) Abstract

A method and apparatus for measuring stereo image disparity, for use in a 3–D modelling system. The method includes processing left and right camera images to form an image pyramid, calculating a disparity map at the coarsest level in the pyramid, and using this disparity map to carry out a warping operation on one of the images at the next–coarsest level, prior to calculating a disparity map for that level. This process is repeated for each subsequent pyramid level, at each level using the disparity map obtained at the previous level for carrying out the warping process, until a final disparity map for the least coarse pair of images in the pyramid is obtained. A computer program product for implementing this method is claimed, as well as a new method and apparatus for calibrating the cameras.



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **AL** | Albania | **ES** | Spain | **LS** | Lesotho | **SI** | Slovenia |
| **AM** | Armenia | **FI** | Finland | **LT** | Lithuania | **SK** | Slovakia |
| **AT** | Austria | **FR** | France | **LU** | Luxembourg | **SN** | Senegal |
| **AU** | Australia | **GA** | Gabon | **LV** | Latvia | **SZ** | Swaziland |
| **AZ** | Azerbaijan | **GB** | United Kingdom | **MC** | Monaco | **TD** | Chad |
| **BA** | Bosnia and Herzegovina | **GE** | Georgia | **MD** | Republic of Moldova | **TG** | Togo |
| **BB** | Barbados | **GH** | Ghana | **MG** | Madagascar | **TJ** | Tajikistan |
| **BE** | Belgium | **GN** | Guinea | **MK** | The former Yugoslav Republic of Macedonia | **TM** | Turkmenistan |
| **BF** | Burkina Faso | **GR** | Greece | | | **TR** | Turkey |
| **BG** | Bulgaria | **HU** | Hungary | **ML** | Mali | **TT** | Trinidad and Tobago |
| **BJ** | Benin | **IE** | Ireland | **MN** | Mongolia | **UA** | Ukraine |
| **BR** | Brazil | **IL** | Israel | **MR** | Mauritania | **UG** | Uganda |
| **BY** | Belarus | **IS** | Iceland | **MW** | Malawi | **US** | United States of America |
| **CA** | Canada | **IT** | Italy | **MX** | Mexico | **UZ** | Uzbekistan |
| **CF** | Central African Republic | **JP** | Japan | **NE** | Niger | **VN** | Viet Nam |
| **CG** | Congo | **KE** | Kenya | **NL** | Netherlands | **YU** | Yugoslavia |
| **CH** | Switzerland | **KG** | Kyrgyzstan | **NO** | Norway | **ZW** | Zimbabwe |
| **CI** | Côte d'Ivoire | **KP** | Democratic People's Republic of Korea | **NZ** | New Zealand | | |
| **CM** | Cameroon | | | **PL** | Poland | | |
| **CN** | China | **KR** | Republic of Korea | **PT** | Portugal | | |
| **CU** | Cuba | **KZ** | Kazakstan | **RO** | Romania | | |
| **CZ** | Czech Republic | **LC** | Saint Lucia | **RU** | Russian Federation | | |
| **DE** | Germany | **LI** | Liechtenstein | **SD** | Sudan | | |
| **DK** | Denmark | **LK** | Sri Lanka | **SE** | Sweden | | |
| **EE** | Estonia | **LR** | Liberia | **SG** | Singapore | | |

# IMPROVED METHODS AND APPARATUS FOR 3-D IMAGING

The present invention relates to apparatus and methods for the acquisition of three-dimensional (3-D) model images using stereo imaging techniques. More specifically, although not exclusively, the invention provides novel calibration and stereo matching methods for use in stereo imaging.

Three-dimensional or "3-D" imaging is of great interest as there are numerous potentially useful applications of such a technique including, for example, capturing 3-D images of objects, human faces or the environment, for quantitative measurement or visual recordal purposes. One known technique involves capturing a pair of stereo images of an object, commonly using two cameras, and "matching" the two images in order to construct a so-called "disparity map" specifying the stereo disparity for each pixel in one image relative to the other image. The disparity map can be used to construct a 3-D model image, from the pair of captured images, for viewing on a computer screen, the 3-D image being rotatable on screen for viewing from apparent different angles.

The disparity map is a two dimensional array which specifies for each pixel p(x,y) in one image (e.g. the left image) of the pair, the displacement {as a vector (x,y)} to a corresponding point $\pi_r$ in the other image (i.e. the right image). By a "corresponding point" in the right image we mean the point in the right image at which the scene component $\pi_s$ imaged by pixel p(x,y) in the left image appears in the right image.

The construction of the disparity map is a crucial step in the 3-D imaging process and the accuracy of the final 3-D model which is obtained will depend on the quality of the disparity map. To date, the matching techniques used to construct such disparity maps have required substantial processing power and computation time, for example around 45 minutes to match two 512x512 pixel images.

Furthermore, one known correlation-based matching method uses a 5 x 5 pixel window to compute correlation between the left and right images. The assumption is that given an image location u = (x,y) in the left image, L, there is a corresponding point u' in the right image, R, such that $L_u = R_{u'} + \Delta_{u'}$ where $\Delta$ is a noise process. The correlation estimate opens a 5 x 5 (in the preferred embodiment) window around u in L and u' in R to determine the most likely u'. To be accurate this formula assumes that there is no disparity gradient over this small window. This assumption is invalid, as in fact the geometric distortion over a 5 x 5 pixel window can be as much as 1 pixel. One solution is to reduce the size of the window, when such distortion occurs. Unfortunately, this increases the effect of noise, and in any case the discrete nature of image sampling cannot eliminate the problem. Another solution is to apply a local affine map. Unfortunately local computation of this map is difficult, and any local fitting process is liable to over-fitting because of the lack of available data.

Another important aspect of the stereo imaging process is the calibration of the cameras used to record the raw images. This usually requires skilled intervention by the operator of the imaging system to carry out a calibration procedure in order

to achieve accurate calibration, or "factory" calibration which reduces the range of operating volumes of the 3-D imaging process. Projective distortion effects in the imaging system can also affect the accuracy of the calibration achieved.

The construction or so-called "recovery" of three-dimensional surfaces from disparity maps, once camera calibration has been achieved, is well understood in the art. Given a number of 3-D models generated from a stereo matching system, or other range-finding device, there are two problems: (a)determine transformations that will bring the different models into registration, and (b)integrate the different 3-D models into a single model. The integration of the 3-D models can be achieved via the popular method of volumetric representation, for example as described by Curless & Levoy (SIGGRAPH 96 Conference Proceedings, pages 303-312). Problems are though often encountered when attempting to smoothly merge photographic render images associated with the individual 3-D models into the integrated 3-D model.

It is an object of the present invention to avoid or minimise one or more of the foregoing disadvantages.

According to a first aspect of the invention we provide a method of measuring stereo image disparity for use in a 3-D modelling system, the method comprising the steps of:

(a) producing a first camera output image of an object scene;

(b) producing a second camera output image of said object scene;

(c) digitising each of said first and second camera output images and storing them in a storage means;

(d) processing said first and second digitised camera output images so as to produce an image pyramid comprising a plurality of successively produced pairs of filtered images, each said pair of filtered images providing one level in the pyramid, each successive pair of filtered images being scaled relative to the pair of filtered images in the previous pyramid level by a predetermined amount and having coarser resolution than the pair of filtered images in said previous pyramid level, and storing these filtered images;

(e) calculating an initial disparity map for the coarsest pair of filtered images in the pyramid by matching one image of said pair of coarsest filtered images with the other image of said coarsest pair of filtered images;

(f) using said initial disparity map to carry out a warping operation on one image of the next-coarsest pair of filtered images in the pyramid, said warping operation producing a shifted version of said one image; and

(g) matching said shifted version of said one image of the next-coarsest pair of images with the other image of said next-coarsest pair of images so as to obtain a respective disparity map for said other image and said shifted image, which respective disparity map is combined with said initial disparity map so as to obtain a new, updated, disparity map for said next-coarsest pair of images; and

(h) repeating steps (f) and (g) for the pair of filtered images in each subsequent pyramid level, at each level using the new, updated disparity map obtained for the previous level as said initial disparity map for carrying out the warping process in step (f), so as to arrive at a final disparity map for the least coarse pair of images in the pyramid.

Said processing step (d) for image pyramid generation may conveniently comprise operating on said first and second digitised camera output images with a scaling and convolution function. The plurality of pairs of filtered images produced by said scaling and convolution function are preferably Difference of Gaussian (DoG) images. Alternatively, the scaling and convolution function may produce a plurality of pairs of Laplacian images. Other filtering functions could alternatively be chosen, as long as each pair of filtered images produced by the chosen function are of a relatively lower resolution than the resolution of the pair of images in the previous pyramid level. The first pair of filtered images produced in step (d) may in fact be of the same scale (i.e. equal in size) as the digitised first and second camera output images. Preferably, each subsequent pyramid level contains images which are scaled by a factor of f, where $0 < f < 1$, relative to the previous level. Preferably, scaling and summing over all levels of the pyramid provides the original digitised first and second camera output images.

Thus, it will be appreciated that step (d) above preferably includes successively producing a scale pyramid of pairs of left and right filtered images, preferably Difference of Gaussian (DoG) images, from said first and second digitised camera output images, each successive scale providing smaller images having a lower ("coarser") resolution. Advantageously, starting with the pair of images of smallest size and lowest resolution (namely the pair of images from the top level of the pyramid), and the initial disparity map obtained therefor, each pair of left and right images in the scale pyramid may successively and sequentially be used to calculate a new, updated, disparity map at the next level down in the pyramid.

This process proceeds from the coarsest to the finest scale of detail, propagating down new, further refined, values for the disparity map for said first and second camera output images at each transition between scales of the pyramid. Preferably, there are at least five levels in the scale pyramid and the process is therefore iterated at least five times. The final disparity map provides an accurate measure of the disparity between the first and second digitised camera output images.

It will be appreciated that the filtered "images" and shifted or "warped" images referred to above are in the form of two-dimensional data arrays. Each disparity map in practice preferably comprises two two-dimensional data arrays, a first said array comprising the horizontal disparity values for each pixel (in a chosen one of said first and second images relative to the other of said first and second images) and a second said array comprising the respective vertical disparity values for each pixel.

A significant advantage of the above-described invention is that the warping operation ensures that the current best estimate of the disparity (for each pixel), inherited from the previous scale or "level" of the pyramid, is always used when matching the pair of images in the next level down in order to calculate a new, even better estimate of the disparity. This tends to minimise the adverse effects of geometric distortion. By iterating this process through all the scale pyramid levels we can obtain a final, very accurate disparity map which can then be used to construct a 3-D or "stereo" image from the original (i.e. first and second) digitised camera output images, as will be described hereinafter.

Optionally, the method may include repeating steps (f) and (g), namely the warping and matching steps, once or more, at one or more of said pyramid levels, using the latest available disparity map for each warping operation. This tends to further improve the accuracy of the final disparity map for the object scene.

The method preferably further includes constructing a confidence map conveniently in the form of a two-dimensional data array, during each iteration of the above-described method. Each confidence map provides an estimate of the confidence with which the calculated new, further refined (horizontal and vertical) disparity values for each pixel is held. The method preferably also includes the further step of carrying out a smoothing operation on the disparity and confidence maps produced at each level in the scale pyramid of images (for said first and second camera output images of the object scene), prior to using these smoothed maps in the calculation of the new, further refined disparity and confidence maps at the next level. The smoothing operation preferably comprises convolving each map with a predetermined weight construction function W(I,a,b,P) which, preferably, is dependent upon the original image intensity (brightness) values, and the confidence values, associated with each pixel. This weight construction function preferably computes a convolution kernel, for example a 3 X 3 kernel, on an image (data array) I at pixel p(a,b) using a probability array P, which is preferably the confidence map C. Advantageously, this convolution is repeated a plurality of times, preferably at least ten, advantageously twenty, times in order to obtain a final smoothed version of each disparity map and confidence map produced at each pyramid level.

The above-described smoothing operation results in a further improvement in the accuracy of the final calculated disparity map which, in turn, results in improved quality in the 3-D image which is ultimately constructed using the first and second digitised camera output images and the final disparity map (together with calibration parameters for the cameras).

In steps (e) and (g) of the above-described method, said matching process by means of which one image is matched with another is preferably carried out in each case by:

(a) calculating horizontal correlation values for the correlation between the neighbourhood around each pixel p(x,y) in one image and the neighbourhoods around at least three horizontally co-linear points in a spatially corresponding area of the second image, and fitting a parabolic curve (namely, a quadratic function) to said horizontal correlation values and analysing said curve (i.e. quadratic function) in order to estimate a horizontal disparity value for the said pixel p(x,y); and

(b) calculating vertical correlation values for the correlation between the neighbourhood around each pixel p(x,y) in one image and the neighbourhoods around at least three vertically co-linear points in a spatially corresponding area of the second image, and fitting a parabolic curve (namely, a quadratic function) to said vertical correlation values and analysing said curve (i.e. quadratic function) in order to estimate a vertical disparity value for the said pixel p(x,y).

The data values (which are real numbers representing image brightness) of the image for fractional points located between

pixels may be obtained by a suitable interpolation method, preferably using bilinear interpolation.

An advantage of this matching method is that the correlation process is carried out at sub-pixel level i.e. correlation values for fractional points located between actual pixels are calculated and used to fit the parabolic curve. This results in a more accurate estimate of the horizontal and vertical disparities being achieved.

For the avoidance of doubt, the terms "horizontal" and "vertical" are used above with reference to the horizontal lines (rows) and vertical lines (columns) of pixels in the images, and not intended to refer to the actual orientation of the image, for example with respect to a surface or horizon.

It will be appreciated that in the above-described stereo matching method we calculate the disparity for each pixel in one of the first and second (digitised) camera images relative to the other of these two camera images e.g. the disparity for each pixel in the first (hereinafter referred to as the "left") image relative to the second (hereinafter referred to as the "right") image - these could be termed the right-to-left disparities. In a further improvement, the entire above-described stereo matching method may be repeated, this time calculating the disparities in the reverse direction i.e. the right-to-left disparities. By comparing the right-to-left disparities with the left-to-right disparities we can detect occluded areas.

For additional accuracy, the method may also include illuminating the object scene with a textured pattern. This

may conveniently be achieved by projecting the pattern onto the object scene using a projector means. Preferably, the pattern comprises a fractal pattern, for example in the form of a digitally generated fractal random pattern of dots of different levels of transparency. Illuminating the object scene with a textured pattern has the advantage of generating detectable visual features on surfaces in the object scene which, due to a uniformity in colour, would otherwise be visually flat. The use of a fractal pattern ensures that texture information will be available in the Difference of Gaussian (DoG) images at all levels in the scale pyramid.

According to a second aspect of the invention we provide a 3-D image modelling system comprising:

first camera imaging means for producing a first camera output image of an object scene;

second camera imaging means for producing a second camera output image of said object scene;

digitising means for digitising each of said first and second camera output images;

storage means for storing said digitised first and second camera output images; and

image processing means programmed to:

(a) process said first and second camera output images so as to produce an image pyramid of pairs of filtered, preferably Difference of Gaussian (DoG), images from said first and second digitised camera output images, each successive level of the pyramid providing smaller images having coarser resolution, and said storage means being capable of also storing the pairs of filtered images so produced;

(b) process the coarsest pair of filtered images in the pyramid so as to: calculate an initial disparity map for said

coarsest pair of filtered images; use said initial disparity map to carry out a warping operation on one said next-coarsest pair of filtered images, said warping operation producing a shifted version of said one of said next-coarsest pair of filtered images; matching said shifted version of said one of said next-coarsest pair of filtered images with the other of said next-coarsest pair of filtered images to obtain a respective disparity map for said other image and said shifted image, which disparity map is combined with said initial disparity map to obtain a new, updated, disparity map for said next-coarsest pair of filtered images;

(c) iterating said warping and matching processes for the pair of images at each subsequent level of the scale pyramid, at each level using the new, updated disparity map from the previous iteration as the "initial" disparity map for carrying out the warping step of the next iteration for the next level, prior to calculating the new, updated, disparity map at this next level, so as to obtain a final disparity map for the least coarse pair (i.e. the finest reolution pair) of filtered images in the image pyramid; and

(d) operating on said first and second digitised camera output images using said final disparity map, in a 3-D model construction process, in order to generate a three-dimensional model from said first and second camera output images.

The 3-D modelling system preferably further includes projector means for projecting a textured pattern onto the object scene, preferably for projecting a fractal random dot pattern onto the object scene.

According to a third aspect of the invention we provide a computer program product comprising:

a computer usable medium having computer readable code means embodied in said medium for carrying out a method of measuring stereo image disparity in a 3-D image modelling system, said computer program product having computer readable code means for:

processing data corresponding to a pair of first and second digitised camera output images of an object scene so as to produce filtered data corresponding to a plurality of successively produces pairs of filtered images, each pair of filtered images providing one level in the pyramid, each pair of filtered images being being scaled relative to the pair of filtered images in the previous level by a predetermined amount and having coarser resolution than the pair of images in said previous level;

calculating an initial disparity map for the coarsest pair of filtered images by matching filtered data of one image of said coarsest pair of filtered images with the filtered data of the other image of said coarsest pair of filtered images;

using said initial disparity map to carry out a warping operation on the data of one image of the next-coarsest pair of filtered images in the pyramid, said warping operation producing a shifted version of said one image; and

matching said shifted version of said one of said next-coarsest pair of images with the other of said next-coarsest pair of images so as to obtain a respective disparity map for said other image and said shifted image, which disparity map is combined with said initial disparity map so as to obtain a new, updated, disparity map for said next-coarsest pair of filtered images; and

iterating said warping and matching processes for the pair of images at each subsequent level of the scale pyramid, at each level using the new, updated disparity map from the previous

iteration as the "initial" disparity map for carrying out the warping step of the next iteration for the next level, prior to calculating the new, updated, disparity map at this next level, so as to obtain a final disparity map for the least coarse pair (i.e. the finest reolution pair) of filtered images in the image pyramid.

The computer program product preferably further includes computer readable code means for:

operating on said first and second digitised camera output image data using said final disparity map, in a 3-D model construction process, in order to generate data corresponding to a three-dimensional image model from said first and second digitised camera output image data.

According to a fourth aspect of the invention we provide a method of calibrating cameras for use in a 3-D modelling system so as to determine external orientation parameters of the cameras relative to a fixed reference frame (one of the said cameras may be used as the fixed reference frame, although this need not always be the case), and determine internal orientation parameters of the cameras, the method comprising the steps of:

(a) providing at least one calibration object having a multiplicity of circular targets marked thereon, wherein said targets lie in a plurality of planes in three dimensional space and are arranged such that they can be individually identified automatically in a camera image of said at least one calibration object showing at least a predetermined number of said circular targets not all of which lie in the same plane;

(b) storing in a memory means of the modelling system the relative spatial locations of the centres of each of said target circles on said at least one calibration object;

(c) capturing a plurality of images of said at least one calibration object with each of a pair of first and second cameras of the modelling system, wherein at least some points, on said at least one calibration object, imaged by one of said cameras are also imaged by the other of said cameras;

(d) analysing said captured images so as to: locate each said circular target on said at least one calibration object which is visible in each said captured image and determine the centre of each such located circular target, preferably with an accuracy of at least 0.05 pixel, most preferably with up to 0.01 pixel accuracy; and identify each located circular target as a known target on said at least one calibration object;

(e) calculating initial estimates of the internal and external orientation parameters of the cameras using the positions determined for the centres of the identified circular targets.

The method preferably includes the further step of:

(f) refining the initial estimates of the internal and external orientation parameters of the cameras using a least squares estimation procedure.

The terms "external orientation parameters" and "internal orientation parameters" of the cameras are well known and understood in the art, and are therefore not described in detail herein. For the avoidance of doubt, the method according to the fourth described aspect of the invention calculates, for each said camera, initial estimates of at least the following internal orientation parameters: the

position (in x-y co-ordinates) of the principal point of the camera; the focal length f of the camera; and the relative size of an image pixel (relative to the x and y axes). The method preferably further includes calculating estimates for further internal orientation parameters, namely lens distortion parameters. The external orientation parameters for which initial estimates are calculated preferably comprise three location (i.e. linear position) parameters and three angular (position) parameters.

Step (e) may conveniently be done using a Direct Linear Transform (DLT) technique.

Step (f) may conveniently be carried out by applying a modified version of the co-linearity constraint, conveniently in a the form of an iterative non-linear least squares method, to the initial estimates of the internal and external orientation parameters of the cameras, to calculate a more accurate model of the internal and external orientation parameters of the cameras, which may include lens distortion parameters.

One advantage of the above-described calibration method is that, by using a calibration object designed to provide a source of points the location of which in space is known accurately, and the location of which can be accurately identified in images, calibration of the cameras can be achieved automatically without user intervention, for example manipulation of the camera positions, being required.

The method preferably also includes the step of modelling perspective distortion, which causes the centres of circular

targets on the or each said calibration object not to be in the centre of the corresponding ellipses appearing in the captured camera images. This process is preferably incorporated in the afore-mentioned iterative non-linear least squares method used to calculate a more accurate model of the internal and external orientation parameters of the cameras, so as to calculate ellipse correction parameters. This enables further improved accuracy to be achieved in the model of the internal and external orientation parameters of the cameras. It will be appreciated that it is the feature of using circular targets in the calibration object(s) which enables such ellipse correction to be achieved.

According to yet another aspect of the present invention we provide a 3-D modelling method incorporating the afore-described method of calibrating cameras, and the afore-described method for measuring stereo image disparity, and wherein the estimated internal and external parameters of the first and second cameras, and the final calculated disparity map for the first and second camera images, are used to construct a 3-D model of the object scene. The 3-D model may be produced in the form of a polygon mesh.

Accordingly, in a yet further aspect of the invention, we provide a 3-D modelling system as afore-described in which the image processing means is further programmed to carry out steps (d) and (e), and preferably also step (f), in the above-described method of calibrating cameras, and wherein the system further includes at least one said calibration object and storage means for storing constructed 3-D models.

Optionally, in any of the above-described 3-D modelling methods and apparatus, more than two cameras may be used, for example a plurality of pairs of left and right cameras may be used, in order to allow simultaneous capture of multiple pairs of images of the object scene. In a preferred embodiment of the 3-D modelling method of the invention, in which multiple pairs of images of the object scene are captured in this manner, each pair of images may be used to produce a 3-D model of the object scene, using one or more of the afore-described techniques, and the plurality of said 3-D models thus produced are preferably combined together in a predetermined manner to produce a single, output 3-D model. This single, output 3-D model is preferably produced in the form of a polygon mesh. Preferably, the plurality of 3-D models are combined in an intermediate 3-D voxel image which may then be triangulated, conveniently using an isosurface extraction method, in order to form the polygon mesh 3-D model.

The method may also include integrating one or more render images onto said polygon mesh so as to provide texturing of the polygon mesh. One or more dedicated cameras may be provided for capturing said one or more render images. In one possible embodiment the 3-D modelling system of the invention may therefore incorporate three cameras, a left and right camera pair for capturing the left and right object scene images, and a third, central camera for capturing at least one image for providing visual render information. Such a camera triple may be referred to as a "pod". The 3-D modelling system of the invention preferably incorporates one or more such pods. Alternatively, in another possible embodiment one of said first and second cameras may be used to capture at least one image for providing visual render.

Preferably, said one or more render images are merged onto the polygon mesh is such a way as to achieve substantially seamless texturing of the polygon mesh, with image blurring kept to a minimum. This may be achieved by using a boundary-based merging technique, rather than an area-based merging approach. Each triangle of the polygon mesh has one or more of said render images projected or mapped thereon, and the system determines which image projects or maps most accurately onto the said triangle by analysing the confidence, or weighting, values associated with the vertices of the said triangle, and preferably also taking into account the size (i.e. area) of the said triangle onto which the render image(s) is/are being projected.

More than one calibration object may sometimes be used in the 3-D modelling method and apparatus of the invention. Where multiple pairs or pods of cameras are used, there is preferably provided a different calibration object for use with each said pair, or each said pod. Each calibration object may advantageously be provided with at least one optically readable bar code pattern which is unique to that calibration object. The image processing means is preferably programmed to locate said at least one bar code pattern for each calibration object imaged by the cameras, and to read and identify said bar code pattern as one of a pre-programmed selection of bar code patterns stored in a memory means of the apparatus, each said stored bar code pattern being associated with a similarly stored set of data corresponding to the respective calibration object. In this manner individual ones of the calibration object may be recognised by the image processing means where multiple calibration object appear in a single captured image.

Each said calibration object is preferably additionally provided with bar code location means in the form of a relatively simple locating pattern which the image processing means can locate relatively easily and, from the location of said locating pattern, identify that portion of the image containing said at least one bar code pattern, prior to reading said bar code pattern.

Preferred embodiments of the invention will now be described by way of example only and with reference to the accompanying drawings in which:

Fig.1 is a schematic diagram of a camera apparatus according to one embodiment of the invention;

Figs.2(a),(b), and (c) show three image frames captured of a human subject, comprising left texture, right texture and left render images respectively;

Fig.3 is a disparity map for left and right captured images;

Fig.4 is a Lambertian shaded view of a 3-D model image of the human subject of Figs.2;

Figs.5(a) and (b) are 3-D mesh models of the human subject;

Fig.6 is the 3-D mesh model of Fig.5(a) overlaid with the left render image of Fig.2(c);

Fig.7 is a random dot pattern;

Fig.8 a stereo pair of images of a shoe last illuminated with the random dot pattern of Fig.7;

Fig.9 shows the stereo image pair of Fig.8 at level five of the scale pyramid obtained therefrom;

Fig.10 shows a fractal random dot image;

Fig.11 shows a stereo pair of images of the shoe last illuminated with the random dot pattern of Fig.10;

Fig.12 shows the stereo image pair of Fig.11 at level five of the scale pyramid obtained therefrom;

Fig.13 a barcode design;

Fig.14 shows front, side and bottom views of a calibration object used in one embodiment of the invention;

Fig.15 is a flow diagram illustrating part of the calibration process used in one embodiment of the invention;

Fig.16 illustrates a contour following set-up;

Figs. 17(a) illustrate the definitions of slant ($\sigma$) angle and tilt ($\tau$) angle respectively;

Fig. 18 illustrates a barcode reading process;

Fig.19 illustrates projective distortion of the center of a circle;

Fig.20 is a flow chart illustrating the formation of a Difference of Gaussian (DoG) pyramid;

Fig.21 is a flow chart illustrating a main loop of a stereo matching process of the invention;

Fig.22 is a flow diagram illustrating a matching process carried out at each level of the scale pyramid; and

Fig.23 is a flow chart illustrating in further detail how the matching process of Fig.22 is used to operate on all levels of the scale pyramid.

The present invention is a 3-D camera apparatus and associated processing methods, capable of capturing 3-D models of a wide variety of objects. Figure 1 shows a 3-D camera apparatus 1 consisting of:

a monochrome or colour left camera 3;

a monochrome or colour right camera 4;

an (optional) central colour camera (not shown in Fig.1);

a calibration object 7 for determining the external and internal orientation parameters of the cameras 3, 4, relative either to the object 7 or to one of the cameras. The cameras 3,4 are, in this embodiment, digital still cameras. However,

in other alternative embodiments the cameras 3,4 may, for example, be videocameras or 35mm cameras.

The calibration object 7 contains an identifying code and a number of circular targets;

a Central Processing Unit 11 (hereinafter referred to as "the CPU" or "the computer") for controlling the operation of the cameras, and projector, and for processing the images to produce a 3-D rendered model;

a digitiser 9 for digitizing each of the output images from the left and right cameras 3, 4 and storing them in the CPU 11;

a projector 13 is controlled by the CPU 11 and, in use, projects a fractal pattern over an object scene imaged by the left and right cameras;

mounting means, such as tripods (not shown), for mounting the cameras 3, 4 so as to allow for user controlled alteration to the separation and orientation of the cameras;

a storage medium (e.g. memory) 15 for storing 3-D models; and two frame buffers 17,18 connected between the digitizer 9 and the CPU 11.

The cameras, imageprocessing hardware, mounting hardware, CPU and storage medium for the present invention are commercially available. Suitable cameras are: left camera Sony XC77-CE, right camera JVC FY55-RGB. Suitable image processing hardware is the DataCell Snapper 24. The separation and rotation of the left and right cameras 3,4 are adjustable, as are the two lenses 6, 8 attached to the respective cameras. This allows a variety of viewing volumes to be imaged. The use of the calibration target 7 (described in detail later), allows rapid

calibration without user intervention after alteration of the positions of the cameras 3, 4 or changes in lenses 6, 8.

A summary of the operation of the apparatus 1 of Fig. 1 will now be given, followed by a more detailed description of the apparatus and methods used.

1. The cameras 3, 4 are calibrated by capturing several images of the calibration object 7, with the object 7 visible from both cameras in at least one of the images. This calibration process is described in more detail later.

2. The projector 13 is set to project a fractal pattern onto the object scene. (Details of the fractal pattern are described later).

3. A target object or human subject to be modeled (not shown) is then placed in front of the camera. In the preferred embodiment for a single pair of cameras using 50mm C-mount lenses the subject is situated at a distance of 1.5 meters from the cameras 3, 4 which are separated by 30cm.

4. The projector 13 is used to light up the subject or target with a textured illumination pattern. This is derived from a 35mm slide with a digitally generated random fractal pattern of dots of different levels of transparency. The purpose of the illumination with textured light is to generate detectable visual features on surfaces which, due to a uniformity in colour, would otherwise be visually flat. (The use of a fractal pattern

ensures that texture information will be available at all levels of a Difference of Gaussian (DoG) image pyramid).

5. The CPU 11 instructs the two cameras 3, 4 to simultaneously capture a single frame of the scene containing the subject.

6. The frames from the left and right cameras 3, 4 are downloaded respectively into the two frame buffers 17, 18. Call these frames $A_f$ and $B_f$. Two such frames $A_f$, $B_f$ obtained of a human subject's head are shown in Figs. 2(a) and (b).

7. The computer instructs the projector 13 to illuminate the subject or target with uniform white light. The purpose of this is to allow the underlying colour and texture of the subject's face, or the target's surfaces to be recorded.

8. One of the cameras, preferably the left camera, is instructed to capture a frame 9 (or "render image")of the target or subject illuminated with white light.

9. The frame $C_f$ is downloaded into the left frame buffer 17. Fig 2(c) shows such a frame captured of the human subject of Figs. 2(a) and (b).

10. The computer calculates the disparity between the left and right images. For each pixel position a horizontal disparity and a vertical disparity) is calculated between the left and right images. The confidence of the disparity estimates, stored as a floating point image, with elements in the range 0-1 is also generated. The disparity and

confidence (in the horizontal and vertical directions) is then organised as two image frames, $D_f$, $E_f$. The horizontal disparity image $D_f$ for the human subject of Fig. 2 is shown in Figure 3. Details of the matching method used to calculate the disparities are given later.

11. This disparity map is translated, using the known internal and external orientation parameters of the left and right cameras 3, 4 into a dense 3-D map or image $F_f$ (shown in Fig. 4), where each pixel in the image $F_f$ represents a position in space. A representation of this map is shown in Fig. 4 where the model has been rendered into a byte-coded image by Lambertian shading of the model, with a light source positioned at the principal point of the left camera.

12. This 3-D model image is translated, using the known internal and external orientation parameters of cameras, into a 3-D mesh representing the subject. This mesh can be conveniently written out as a VRML file on the storage medium 15. Figs. 6 (a) and (b) show the mesh from two different angles of view. The fact that the mesh is a 3D model allows the angle of view to be altered so that a computer can present the same object from alternative angles.

13. The CPU 11 stores on the storage medium 15, along with the 3D mesh, Frame $C_f$, the image obtained from the left camera with white light. With appropriate graphics software it is possible to reconstruct different fully textured views of the original subject, such as that shown in fig. 6, using the stored 3-D mesh and the render image (frame $C_f$).

In the preferred embodiment more than two cameras are used. These are described herebelow. The preferred operation of the system, when more than two cameras are used organises the cameras in triples, with left and right monochrome cameras and a central color camera in each triple. We shall call each such camera triple a pod. The left and right cameras 3, 4 form a stereo pair for matching purposes, while the central camera is used to capture the visual render image Cf. The cameras are connected in a tree structure via a Universal Serial Bus (USB) (not shown), using USB hubs to form the connection topology.

Multiple projectors 13 may be used, and the projectors are positioned so that, the fractal texture pattern projected thereby covers the object or objects to be captured. The design of the fractal texture is such that the output of multiple projectors may overlap.

Multiple calibration objects 7 may be used. The design of the objects 7 is such that multiple calibration objects may be recognised when they occur in a single image, and identified. The design of the objects and the methods employed are described in full detail herebelow. Where multiple calibration objects 7 are used, operation proceeds as follows:

1. The cameras are calibrated by capturing multiple images of the calibration objects. In order to ensure that all cameras are calibrated within the same co-ordinate frame the following constraints must be satisfied. Define a graph $\langle V,E\rangle$ with edges E and vertices V. The vertices are $C \cup O$ where C is the set of cameras, and O is the set of calibration objects being used. An edge $E = \langle c,o\rangle, c \in C, o \in O$ is in the graph if there exists an image I captured by camera c with o visible in

the image. Similarly an edge $E = \langle o,c \rangle, c \in C, o \in O$ is in the graph if there exists an image I that contains o and I was captured by camera c. The constraint that must hold for calibration to succeed is that the graph V is connected. (Details of the calibration method can be found in a later section of this description).

2. Each projector is set to project the fractal pattern onto the subject. (Details of the fractal pattern are provided later).

3. A target object or human subject is placed in front of the cameras.

4. The computer issues a signal to the left and right camera pairs in the pods via the USB, instructing them to capture an image. The captured imaged is stored on internal memory in each camera.

5. The computer instructs the central cameras on the pods to fire simultaneously, via the USB, and store the resulting image. These pictures are captured in ambient light. For optimal texture the ambient light should be controlled to achieve the illumination desired, as with a conventional film camera. This can conveniently be done using standard photographic flash equipment.

6. The frames from each pod are transferred under control of the CPU via the USB, either to the storage medium 15 or to the computer memory.

7. For each pair of left and right cameras, the computer calculates the disparity between the left and right images (as described later).

8. The disparity maps are used together with the known internal and external orientation parameters of the left and right cameras to generate dense 3-D maps as described in above.

9. The dense 3-D maps, together with the internal and external orientation parameters of the left cameras, are used to generate a 3-D implicit surface image of the complete region imaged. This process is explained in detail later.

10. The implicit surface is polygonized, using a method such as Marching Cubes [Lorensen and Cline, SIGGRAPH '87 Conference proceedings, Anaheim,CA, July 1987, p.163-170); Mesh Propagation [Howie and Blake, Computer Graphics Forum, 13(3):C/65-C/74, October 1994]; or span space methods [Livnat et al., IEEE Transactions on Visualisation and Computer Graphics,2(1):73-84, March 1996], to generate a triangular mesh. The triangle mesh can be decimated by applying, for example, the method of Garland and Heckbert (SIGGRAPH 96 Conference Proceedings, Annual Conference Series, p.209-216, 1997] if required for ease of display.

11. The white light (render) images from the center cameras, the internal and external orientation parameters of the center cameras, and the location in space of the vertices of the generated polygon mesh are used to merge the render images, so that they can be projected seamlessly onto the polygon mesh. This method is also described later.

12. If accurate measurement from the generated models is required, the polygon mesh can be used to merge the dense models, using the same method as in 11 above, and can also be used to mask the areas of the models which correspond to the object(s) being imaged.

13. The computer 11 stores along with the 3D mesh on the storage medium 15, the images obtained from the central cameras and their internal and external orientation parameters. If accurate measurements and the highest quality rendered images of the object(s) from different

viewpoints are required, the merged dense 3-D model images obtained in step 12 above can also be stored in the storage medium 15.

**FRACTAL TEXTURE PATTERN**

Images of objects illuminated with natural light can contain visually flat areas where accurate estimation of disparity is difficult. When maximum accuracy is required the current invention uses texture projection to illuminate the object(s) of interest with a textured pattern which provides information for the stereo matching method (described later). One way to do this is to project a random dot pattern onto the object to be modeled. A random dot pattern can be generated as follows.

1. Initialise a byte-coded image, arranged as an array of numbers in memory. This image should have a user-defined size. In the preferred implementation this is 600 X 800.
2. Initialise a random number generator, which generates uniformly distributed random numbers in the range 0-255 (the range of a byte).
3. For each pixel of the image, assign a new random value generated by the random number generator.

Fig. 7 shows a random dot pattern and Fig. 8 shows a stereo image pair of a shoe last illuminated with the random dot pattern. Fig. 7 is a non-fractal image. The match method (described later) estimates disparities at each level of a DoG pyramid created from a stereo image pair. The fine grain of the random dot pattern gradually diminishes towards the top of the pyramid, as illustrated in Fig. 9 which shows the stereo image pair of fig. 8 at level 5 of the pyramid (the original image is taken to be level 0).

To eliminate this problem of diminishing pattern, in the method and apparatus of the present invention we use a fractal random dot pattern, which maintains the same image properties at all levels of the DoG pyramid. An example is shown first and then the method to generate the fractal random dot image is described. Fig. 10 shows a fractal random dot image and Fig. 11 shows a stereo image of the same cast illuminated with the fractal random dot image projection. Fig. 12 shows the stereo image pair of Fig. 11 at level 5 of the pyramid. Comparing these Figs. with the respective ones of Figs.7 to 9, as seen from Fig.10, the fractal pattern is in general slightly more dense in some areas than the random dot pattern of Fig.7. Comparing Fig.11 with Fig. 8 it can be seen that the fractal dot pattern is preserved better (throughout the pyramid), at level 5 of the pyramid, than the non-fractal pattern of Fig. 7.

The method used in the current invention to generate the fractal image is:

1. create a stack of random dot images, $r_\iota(x_\iota,y_\iota)$, r = 0,1,...,255; $\iota$ = 0,1,...L-1; $x_\iota$ = 0,1,...,$X_\iota$-1; $y_\iota$ = 0,1,...,$Y_\iota$-1. The ratios of $X_\iota$ : $X_{\iota+1}$ and $Y_\iota$ : $Y_{\iota+1}$ are the same, and are consistent with the ratio between pyramid levels of the matcher. It is 2 for the preferred embodiment of the stereo matching method.
2. each of these images is enlarged by pixel repetition to the same size as the largest one $X_0$ x $Y_0$: $r'_\iota(x, y) = r_\iota(x_\iota, y_\iota)$, $x=x_\iota 2^\iota, x_\iota 2^\iota+1, \ldots, x_\iota 2^{\iota+1}-1$, $y = y_\iota 2^\iota, y_\iota 2^\iota+1, \ldots, y_\iota 2_\iota^{+1}-1$.
3. the average of $r'_\iota(x, y)$ is taken as the fractal random dot image, $r(x,y) = 1/L\ [\sum^{L-1}_{\iota=0} r_\iota(x,y)]$.

**CAMERA APPARATUS**

Suitable cameras for the preferred embodiment with two cameras are: left camera, Sony XC77-CE, right camera, JVC FY55-RGB. When multiple cameras are used, conventional cameras of this type are less suitable. In addition, the requirement for many cameras requires use of a cheaper system with some additional capability, in particular the ability for a subset of the cameras to capture images simultaneously. The preferred embodiment for multi-camera systems therefore uses cameras with the following functionality:

1. Progressive scan.
2. Ability to fire anychronously, to enable multiple cameras to fire simultaneously.
3. Ability to store images in local memory, for progressive download over serial (USB) bus.
4. USB connection supplying both power and control.

The preferred embodiment, uses a VVL 5850 CMOS sensor, and a Hitachi 3048G processor, with 16MB of on-board memory. The camera is connected via USB to the host computer, thereby allowing at least 32 cameras to be connected to a single computer, and each camera to be powered over the USB.

**THE CALIBRATION OBJECT**

It is an objective of the system that accurate camera calibration can be obtained automatically. A simple way to determine camera calibration is to use the position of known points in the world, which are imaged by the camera. If lens distortion is to be modeled accurately a large number of points, whose position is known with precision are required.

The calibration target is designed to provide a source of points the location of which in space is known accurately, and the location of which can be determined accurately in images.

In order for several cameras to be calibrated within the same world co-ordinate framework, it is necessary that some points imaged by a camera are common with at least one other camera. The calibration target is designed in such a way that it can be recognised automatically from any orientation, and several objects can be present in the field of view at any time and each be individually recognised. This enables many cameras to be calibrated together, which is necessary when more than 1 pair (or triple) of cameras is used.

The calibration object is illustrated in Figs. 13 (a), (b) and (c). The preferred embodiment of the calibration object includes two main features. Feature one is a barcode and feature two is a set of targets. The barcode consists of a barcode section and two anchor points on each side of the barcode section. Figure 14 shows the barcode scheme. The set of targets consists of twelve circles 30 lying in two planes 32,33 in 3-D space. (In other possible embodiments only 8 target circles are used). The centre of each circle is a "target point". The arrangement of the twelve circles in two planes is such that the calibration object can be recognised by a camera positioned at any position within an angular viewing range of approximately 60 degrees (subtended from the object plane containing the calibration object), said angular viewing range being in a plane perpendicular to the object plane, and centered on an axis extending perpedicularly to the object plane.

The barcode comprises 6 digits and therefore handles up to 64 calibration objects. When that number is exceeded, some mechanism needs to be put in place to handle that. It is straightforward to extend from 6 digits to 18 digits (up to 262144 calibration objects) and still maintain backward compatibility by adding one barcode sequence on top of the current one and one below.

The data of the calibration object is $o_i = \langle (id,x,y,z,d) \mid id \in 0,1,\ldots,no_i \rangle$, where d is the diameter of the circular target. This data is built into the program. The data are design data, not measured data. The program works for any calibration object of this configuration up to a scale factor.

The target finder has three tasks to carry out. These are to identify the calibration object (if more than one calibration object is used), as each one is different from the others; to find and measure the target points and to identify the targets. To identify the calibration object, the barcode anchor points, namely the 2 con-circles 35,36 on either side of the barcode, are identified. From their positions, the portion of the photograph that holds the barcode is outlined and the barcode is read.

To find and measure the target points, the contours of the circles are followed, and their centers are calculated.

To identify the individual targets, the configuration of the photograph target points is matched with projection of calibration object target points. Since at this stage the camera parameters are unknown, a search of the projection is carried out to find one projection that gives the best match.

Figure 15 is a flow chart illustrating the various steps carried out in the above process which will now be described in further detail.

**IMPLEMENTATION**

For each photograph or image I(i,j),i-0,1,...,M-1; j = 0,1,...,N-1, the target finder does the following:

CONTOUR FOLLOWING

The purpose of the contour following routine is to find all contours in the image. A contour is defined as a list of closed and connected zero crossings. Closed means that the last zero crossing in the list is the same as the first zero crossing of the list. Connected means that their positions are not more than $\sqrt{2}$ pixels apart. A zero crossing is defined as a position in an image which has one of the 4 configurations:

a) a negative pixel on its left side and positive pixel on its right side;

b) a positive pixel on its left side and negative pixel on its right side;

c) a negative pixel on its top side and positive pixel on its bottom side;

d) a positive pixel on its bottom side and negative pixel on its top side.

(It will be appreciated that the pixel values, which represent pixel brightness, in the various DoG images are real numbers i.e. can be "positive" or "negative", as referred to in (a) to (d) above.)

The contour following process uses the setup shown in Figure 16. The current pixel is d(i,j). A contour enters through one of 4 sides UP, DOWN, LEFT or RIGHT. It exits through one of the other sides, and the adjacent pixel on that side is then tracked. If the contour exits d(i,j) through side s it enters the adjacent pixel through next(s), where the function next implements the following mapping:

UP → DOWN, DOWN → UP, LEFT → RIGHT, RIGHT → LEFT

The adjacent pixel to d(i,j) for a given side is determined by the following mapping (adjacent): UP → d(i-1,j), DOWN → d(i+1,j), LEFT → d(i,j-1), RIGHT → d(i,j+1). If any of these pixels lies outside the bounds of the image, then the adjacent pixel is NONE.

1. The input of the routine is an image, I(i,j).

2. The output of this routine is a list of contours, $X = \langle C^l, l = 0,1,\ldots L-1\rangle$, where $C^l = \langle (x_n^l, y_n^l, z_n^l), n = 0,\ldots,L_l-1\rangle$, where x and y provide the image position of a zero crossing and z the product of the 2 pixels that define the zero crossing. Note that z is always negative, and the magnitude of z indicates the strength of the zero crossing.

3. It creates a binary registration image R(i,j), i = 0,...,M-1; j = 0,...,N-1. Each pixel of R, r(i,j), has the value of either processed or unprocessed, and is initialised to unprocessed.

4. It filters the image I with a DoG filter to get $I_{d(\sigma)}(i,j)$ such that a transition from either black to white or vice versa in I becomes a zero crossing in $I_{d(\sigma)}(i,j)$, where σ is

the parameter of positive number that controls the shape of the DoG filter. The parameter σ is chosen to be as small as possible to maximise zero crossing accuracy and at the same time to be great enough to suppress noise in the image. See Figure 4 for an example.

5. For each pixel d(i,j) in $I_{d(\sigma)}(i,j)$, start to trace if a zero crossing is found.
   (a) Set last to NONE.
   (b) If r(i,j) is *processed* return failure.
   (c) Create a temporary empty contour, $C^1$.
   (d) Set current_pixel to (i,j).
   (e) For each side s ≠ last if there is a zero-crossing at s then calculate the precise location of the zero crossing by linear interpolation and add it to $C^1$. Set last to next(s) and set the current pixel to adjacent(current_pixel). If the current pixel is NONE return failure. If the current pixel is equal to the starting pixel d(i,j) return success and add C(1) to the set of contours. Otherwise goto 5(e).

ELLIPSE COMPUTATION

Given a contour $C = \langle (x_i, y_i, z_i),\ i = 0,\ldots,L_c-1 \rangle$ where $x_i, y_i$ is the position of the zero crossing in floating point co-ordinates and z is a negative floating point number that indicates the strength of the zero crossing, we can determine an ellipse with the following parameters:

**zeroz** The average strength of the zero crossings, defined as

$$zerox = \frac{-1}{L_c}\left\{\sum_{n=0}^{L_c-1} z_n\right\}$$

**polarity** Polarity is a value that indirectly describes whether a circle in the image is black on a white background or the other way round. When an image that contains white ellipses on black background is filtered by a DoG filter, the inside of the contours of zero crossings that correspond to the edges of the ellipses can be either negative or positive. But it is consistent for a DoG filter, say negative. When the DoG filter is negated, it will be positive. Polarity takes binary values, positive-inside or negative-inside, to account for this.

**length** Defined as

$$length = \sum_{n=0}^{L_c-1} \left| \{x_n, y_n\} - \{x_{(n-1) \bmod L_c}, y_{(n-1) \bmod L_c}\} \right|$$

**centre** Defined as

$$centre = \frac{1}{length} \left\{ \sum_{n=0}^{L_c-1} \{x_n + x_{(n-1) \bmod L_c}, y_n + y_{(n-1) \bmod L_c}\} \{x_n, y_n\} - \{x_{(n-1) \bmod L_c}, y_{(n-1) \bmod L_c}\} \right\}$$

**mean radius** Defined as

$$meanr = \frac{1}{L_c} \left\{ \sum_{n=0}^{L_c-1} \left| \{x_n, y_n\} - centre \right| \right\}$$

**circularity** Defined as

$$variance = \sqrt{\frac{1}{L_c} \left\{ \sum_{n=0}^{L_c-1} \left( \left| \{x_n, y_n\} - centre \right| - meanr \right)^2 \right\}}$$

$$circularity = \frac{meanr}{variance}$$

**minimum radius, maximum radius, orientation** These three parameters are determined by a least squares fit of the data

points to an ellipse. This is done using the following process:

1. Centralise the contour. $(x_n, y_n) \leftarrow (x_n, y_n)$ - center, $n = 0, \ldots, L_c - 1$.

2. We can express the ellipse in the form $ax^2 + 2bxy + cy^2 = 1$ since we assume that the ellipse is centered on the origin. Minimise the expression $E = \sum_{n=0}^{L_c-1} (ax_n^2 + 2bx_ny_n + cy_n^2 - 1)^2$ by a linear least squares method.

3. In matrix form the ellipse equation is

$$\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1$$

The matrix $P = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ is symmetric and therefore there is a decomposition $T^T \Lambda T = P$

where $T = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$, $TT^T = I$ and $\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$, $\lambda_1 \geq \lambda_2$.

4. The maximum radius ($r_{max}$) is $\lambda_1$, the minimum radius ($r_{min}$) is $\lambda_2$ and the orientation ($\theta$) is arctan (-s/c).

*fit* The goodness of fit of the calculated ellipse to the data points is given by

$$fit = \sqrt{\frac{1}{L_c}\sum_{n=0}^{L_c-1}(rx_n^2 + ry_n^2 - 1)^2}$$

where $(rx_n, ry_n) = SR((x_n, y_n) - centre)$,

$$S = \begin{pmatrix} 1/r_{max} & 0 \\ 0 & 1/r_{min} \end{pmatrix}, \quad R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

The above parameters are used to filter the contour set. In the preferred implementation $r_{min}$, *fit* and zerox are used. When any of these parameters lies outside specified bounds the contour is ignored.

Using the above technique we have been able to determine the centers of the circular targets on the calibration object to within 0.02 of a pixel for a circle with a radius of 30 pixels in the image.

**CONCENTRIC CIRCLE DETECTION AND POINT GROUPING**

The purpose of this routine is to organise the points into barcode anchor point group and target point group. In case of an image of multiple calibration objects, a further organisation is carried out to group together the group of barcode anchor points and the group of target points that belong to the same calibration object.

The input of this routine is a set of ellipses, $E = \langle E_i \rangle$ 1-0,...,$N_i$-1, and the output is a set of group pairs, where each pair consists of a group of barcode anchor points and a group of target points. Namely, $\Gamma = \langle G_b^l, G_t^l \rangle$, $l = 0,...,G-1$, where $G_b^l = \langle E_{in} \rangle$ n=0,1 and $G_t^l = \langle E_{in} \rangle$, $n = 0,...,T_i-1$.

1. Determine the concentric circles by determining whether the centers of pairs of ellipses co-coincide.
2. Divide the concentric circles into groups by pairing them off, since the number of barcode anchor points for each calibration object is 2.
3. Divide the remaining ellipses that are not concentric circles into groups by comparing their distance from the barcode anchor points.

**POINT CONFIGURATION MATCHING**

The purpose of this routine is to match the points of the calibration object $o_i$, $\langle (id,x,y,z,d) \mid id \in 0,1,...,n_{oi} \rangle$, with the image points, namely, $G = \langle G_b, G_t \rangle$ where $G_b = \langle E_n \rangle$, $n = 0,1$ and $G_t = \langle E_n \rangle$ $n = 0,...T-1$. When done, every point in the image has an id assigned.

1. Pre-processing image points

(a) Barcode point:

i. Get point $p_b^n = G_b.E^n.centre$, $i = 0,1$.

ii. Negate y-axis: $p_b^n.y = -p_b^n.y$, $i = 0,1$.

(b) Target point:

i. $p_t^n = G_t.E^n.centre$, $i = 0,...,T-1$.

ii. Negate y-axis: $p_t^n.y = -p_t^n.y$, $i = 0,..., T-1$.

(a) Center calibration object points. Let the points of calibration object o be $p^0,\ldots,p^{no}$. Then:

$$center = \frac{1}{n_o}\sum_{i=0}^{n_o-1}(p_x^i, p_y^i, p_z^i)$$

$$(p_x^i, p_y^i, p_z^i) \leftarrow (p_x^i, p_y^i, p_z^i) - center$$

(b) Extract the target points and the anchor points. The anchor points are those with concentric circles.

3. Find the best match between the calibration object target points and image target points.

(a) Find the best slant angle σ of the calibration object (The slant angle σ is illustrated by Figure 17(a)).

$$\cos(\sigma) = \frac{\sum_{i=0}^{T-1} G_{t_i}.r_{\min}}{\sum_{i=0}^{T-1} G_{t_i}.r_{\max}}$$

(b) Find the best tilt angle τ (illustrated in Figure 17(b)) of the calibration object. This is done via an exhaustive search with respect to the anchor points and the tilt angle. For the two possible ordering of the anchor points and for a number of angles that span the range of 0 to $2\pi$, a match is made between the orthogonal projection in the z-axis of the slant and tilt rotated calibration object points and the affine transformed image target points, where the affine transform is determined by the mapping between the rotated anchor points and the image barcode points. The combination of the tilt angle and the ordering of anchor points that gives the best match is chosen as the true value. To avoid occlusion

the tilt angle and the ordering of anchor points that gives the best match is chosen as the true value. To avoid occlusion problems, the calibration object points that lie behind other points in the orthogonal projection are detected and ignored. The diameter of a circle in the calibration object is used to do the detection. After rotation, an occluded point is found when the projection of the point lies inside the radius of another point that is closer to the image plane. To do a match of a particular tilt angle and ordering of anchor points, the following is carried out.

(i) Rotate the anchor and target points. Ignore occluded points.

(ii) Determine the affine transform, and carry out the transform on the image target points.

(iii) Match the transformed calibration object target points and the image target points using a least squares estimator. Let the k-th image point be $t^i_k$ and the k-th transformed object point be $t^o_k$ then the error estimate is:

$$e = \sqrt{\frac{1}{T}\sum_{k=0}^{T-1}(t^i_k - \min_{\forall j t^o_j}((t^o_j - t^i_k)^2)^2}$$

4. Assign *ids* to image target points

(a) Accept the best match.

(b) Assign to each image target point the *id* of a calibration object target point that is closest.

(c) When an image target point has more than one *id*, those *ids* whose associated calibration object points are further away are ignored.

(d) Assign to each image barcode point the *id* of the calibration object anchor point that is closest. This is to establish the direction of barcode reading.

**BARCODE READING**

The purpose of this routine is to find a parallelogram in the image that contains the barcode. After the ids of the image points are identified, the geometrical relations between points are identified. For instance, the line passing though point 5 and point 9 and the line passing through point 6 and point 10 approximate the direction of the vertical sides of a parallelogram. The two barcode points approximate the direction of the horizontal sides of the parallelogram. The actual positions are approximated through the known positions of barcode points and known sizes of barcode point circles. The parallelogram is represented as 4 points, $(t_l, t_r, b_l, b_r)$, the image pixel locations of the top left corner, the top right corner, the bottom left corner and the bottom right corner of the parallelogram.

The purpose of this routine is to read the 6 digit barcode of a calibration object. It reads evenly spaced lines parallel to the horizontal sides of the parallelogram starting from the left vertical side to the right vertical side of the parallelogram. Then it combines the readings and converts them into an integer. Figure 18 shows an example of reading number 4 in operation. The operation of the reader is as follows:

1. Sample 225 pixel values along five lines in the parallelogram.

2. 2.Determine a threshold between black and white on the barcode by histogramming, and threshold these lines.
3. Determine edges along the 5 sample lines, and convert to a binary number.
4. Convert to a decimal barcode, and determine the object identity.

**CALIBRATION**

Calibration is achieved using a novel version of the colinearity constraint as described by C.C. Slama in "Manual of Photogrammetry, fourth addition, American Society of Photogrammetry and Remote Sensing, Virginia, 1980. This allows a large number of accurately measured points to be used to determine the orientation and internal parameters of the cameras.

In the preferred implementation calibration proceeds as follows:

1. A number of image pairs (or triples if 3 cameras are used) of the calibration object are captured. To determine the cameras' external orientation parameters, and the internal parameters excluding lens distortion, 1 pair (or triple) of images is sufficient. To accurately model lens distortion a larger number of images, up to 30, is required.
2. The location and position of the centers of the target circles on the calibration object are determined for each image.
3. An initial estimate of the position and orientation parameters for the calibration objects, and the internal and external orientation parameters of the cameras is obtained using a version of the DLT algorithm.

4. By use of a modified version of the colinearity constraint, described below, a precise estimate of the parameters is obtained from the initial estimate, using a non-linear least squares method.

**THE MODIFIED COLINEARITY CONSTRAINT**

The colinearity constraint in our preferred implementation is now described. The basic equation is:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \lambda \mathrm{M} \begin{pmatrix} u + \Delta u - u_c \\ v + \Delta v - v_c \\ f \end{pmatrix}$$

where M is the rotation matrix of the camera w.r.t. the default world coordinate frame; $(t_x, t_y, t_z)^T$ is the location of the camera's principal point; $(x,y,z)^T$ is the location of a point in the world (w.r.t. the default co-ordinate frame); $(u,v)^T$ is the location in the image plane, w.r.t. the camera coordinate system, of the image of $(x,y,z)^T$; $\Delta u$ and $\Delta v$ are corrections to the values of u and v due to lens distortion and the distortion due to the observed center of a calibration circle in the image not being the image of the circle center in space; f is the focal length of the camera, and $(u_c, v_c)^T$ is the location, in camera space, of the principal point.

This equation differs somewhat from the standard equation in that the location of the world co-ordinate is on the LHS of the equation. Our preferred implementation extends this equation to take account of the fact that we are taking many

images of the calibration object at different orientations, so that all the points on a given pair (or triple) of images of the calibration object share a common reference frame which is not the default world frame. The revised equation is:

$$\mathbf{M}^o \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \mathbf{T}^o - \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \lambda \mathbf{M} \begin{pmatrix} u - u_c \\ v - v_c \\ f \end{pmatrix}$$

where $M^\circ$ and $T^\circ$ are the rotation matrix and translation for the object reference frame.

This equation is converted into a constraint by dividing through by the z-component to eliminate $\lambda$. This gives us:

$$\begin{aligned} \frac{m^o_{00}x + m^o_{01}y + m^o_{02}z + t^o_x - t_x}{m^o_{20}x + m^o_{21}y + m^o_{22}z + t^o_z - t_z} - \frac{m_{00}(u + \Delta u - u_c) + m_{01}(v + \Delta v - v_c) - m_{02}f}{m_{20}(u + \Delta u - u_c) + m_{21}(v + \Delta v - v_c) - m_{22}f} &= 0 \\ \frac{m^o_{10}x + m^o_{11}y + m^o_{12}z + t^o_y - t_y}{m^o_{20}x + m^o_{21}y + m^o_{22}z + t^o_z - t_z} - \frac{m_{10}(u + \Delta u - u_c) + m_{11}(v + \Delta v - v_c) - m_{12}f}{m_{20}(u + \Delta u - u_c) + m_{21}(v + \Delta v - v_c) - m_{22}f} &= 0 \end{aligned}$$

The parameters in these equations, the values of which are to be determined are:

$(r^\circ_x, r^\circ_y, r^\circ_z)$ The Eulerian rotation angles for the calibration object w.r.t. the default world co-ordinate frame.

$(t^\circ_x, t^\circ_y, t^\circ_z)$ The translation for the calibration object w.r.t. the default world co-ordinate frame.

$(r_x, r_y, r_z)$ The Eulerian rotation angles for the camera w.r.t. the default world co-ordinate frame.

$(t_x, t_y, t_z)$ The translation for the camera w.r.t. the default world co-ordinate frame.

$(u_c, v_c, f)$ The principal point of the camera w.r.t. the camera frame. $u_c$ and $v_c$ are in pixel co-ordinates. The translation from camera space to pixel coordinates involves:

$x_p$ The size of an image pixel along the x-axis.

s The ratio between an x-pixel and y-pixel.

$(k_1,k_2,k_3,k_4,k_5)$ Distortion parameters for the camera.

$o_r$ The radius of the target circle, the center of which is at $(x,y,z)^T$. This is necessary to calculate the ellipse distortion.

It should be noted that the expressions $\Delta u$ and $\Delta v$ are functions of the distortion parameters, the rotation and translation parameters for both the object and the camera, and the radius of the target.

Given a pair of cameras and multiple image pairs of a calibration object captured from the object calibration proceeds in two steps.

1. An initial estimate of the camera parameters, both internal and external, together with transformation parameters for the positions of the calibration object in the various image pairs is derived.

2. Given this initial estimate a non-linear iterative least squares method is used to refine the estimate and achieve precise calibration. At this stage lens distortion and ellipse correction parameters are derived, following to the model described above.

The initial estimate of camera and target parameters is derived using the Direct Linear Transform (DLT), the preferred implementaiton of which is described below.

**INITIAL PARAMETER ESTIMATION USING THE DLT**

The DLT provides an estimate of camera orientation and internal parameters w.r.t. a set of world points. To achieve an estimate of the orientation of multiple cameras and the calibration target in various orientations requires several applications of the DLT. The algorithm for initial parameter estimation takes as input a set of image pairs (triples) of the calibration target with target points identified and located in each image. The DLT implementation given s set of points provides an estimate of the camera parameters, relative to the co-ordinate of the calibration object. The internal parameters of the camera, are of course independent of the co-ordinate frame. A C++ pseudo-code version of the algorithm is given below:

```
void initial_estimate(set<imagepoints> images, set<cameras> cameras)
{
  <set all cameras to undone>
  <set all images to undone>
  <set first image's translation and rotation to 0>
  <set first image as done>
  while <exists camera that is not done> {
    <foreach done image i> {
      if <exists undone camera for i> {
        <set camera from i using the DLT>
      }
    }
```

```
    <foreach done camera c> {
      <foreach undone image i for c> {
        <calculate the DLT for camera c' from i> (*)
        <apply the inverse of this transform to determine the orientation of i>
        <set i done>
      }
    }
  }
}
```

The step in this algorithm labelled with an asterisk determines the orientation of an object with respect to a camera with known orientation. Let a camera c have external orientation M, expressed as a homogenous 4 x 4 matrix, which includes both the rotation and translation parameters. If the DLT is run and determines a matrix $M_o$ for the camera relative to the object in the default world frame, then the orientation for the object relative to camera c will be $M M_o^{-1}$. From this matrix it is possible to recover rotation and translation parameters. On termination of this algorithm, we have an initial estimate for the internal and external orientation parameters of both (all three) cameras and the orientation and location of the co-ordinate system for each image object.

**REFINEMENT OF THE ESTIMATE USING NON-LINEAR LEAST SQUARES**

Given an initial estimate of parameters obtained by the above algorithm, a least squares refinement process is used to provide a more accurate estimate of these parameters, of the lens distortion parameters for each camera, and of the elliptical correction for each object point in each image. The least squares problem is set up as follows:

1. For each point on each image of the calibration object, two constraints can be created, as detailed above in the description of the modified co-linearity constraint.
2. For each point on the calibration object three constraints can be applied specifying the x, y, and z location of the points w.r.t. the default world coordinate frame.

The total number of parameters to be solved for is: $15n_c + 6(n_i-1) + 12 \times 3$, where $n_c$ is the number of cameras, $n_i$ is the number of images of the calibration object taken by the cameras (at least 1 for each camera), and 12 is the number of target points on the calibration object. The parameters for each camera break down as: 6 for external orientation, 3 for internal orientation, 1 to specify the ratio between pixel sizes along the x and y dimensions, and 5 for lens distortion.

The number of constraints available is $2 n_i p_i + 3 \times 12$, where $p_i$ is the number of circular targets detected in the ith image, which is at most 12.

In practice it is often satisfactory to use only one lens distortion parameter, in which case one image from each of two cameras is sufficient.

The Levenberg-Marquand algorithm (described later) is used to solve this problem. In our preferred implementation an algorithm such as that proposed by J J Moré (in G A Watson, Lecture notes in mathematics 630, pages 105-116, Berlin, 1928, published by Springer-Verlag) or J E Dennis, D M Gray, R E Welsh (Algorithem 573 NL2SOL : An Adaptive non-linear last squares algorithm, ACM Transaction on Mathematical software, 7:369-383,1981) should be used which can approximate the

Jacobian. The reason for this is that partial derivatives are hard to calculate for the ellipse correction, which depends on several other parameters. No degradation in the performance of the method when using a finite differencing approximation to the Jacobian has been observed.

**THE PREFERRED EMBODIMENT OF THE DLT ALGORITHM**

Photogrammetry in our system requires the solution of non-linear least squares equations. A major problem is the determination of a starting position for the iterative solution procedure.

A basic photogrammetric problem is to determine a camera's internal and external parameters given a set of known world points which are imaged in the camera. The standard method used by C.C. Slama (referenced above) is to use the colinearity constraint to set up a series of non-linear equations which, given a suitable starting point can be iterated to get an accurate solution to the cameras internal and external orientation parameters.

To determine a suitable starting point the Direct Linear Transform (DLT) introduced in Y F Abed-Aziz and N M Karara ("Direct linear transformation from comparator co-ordinates into object co-ordinates in close range photogrammetry; Proceedings of the ASP Symposium on Close-Range Photogrammetry, pages 1-18, Illinois, 1971) is the preferred method used in the present invention. This method is now described.

The method uses a modified pinhole model for cameras. The modification being that some allowance is made for lens

distortion in the system. Lens distortion is not used in the DLT, although versions have been developed that make allowance for it. The basic equation relating the image coordinates to the world position of the camera is

$$\begin{pmatrix} ux_p & - & p_x \\ -vsx_p & - & p_y \\ & & -p_z \end{pmatrix} = \lambda \begin{pmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{pmatrix} \begin{pmatrix} X - t_x \\ Y - t_y \\ Z - t_z \end{pmatrix}$$

where the components $r_{ij}$ form a rotation matrix, representing the rotation of the camera. Our system uses a right handed coordinate frame, with the camera by default pointing down the -ve z axis. A rotation vector (w,p,κ) represents a rotation of ω radians about the x-axis, followed by a rotation of p radians about the y-axis and finally a rotation of κ radians about the z-axis. The matrix for this rotation is:

$$\begin{pmatrix} \cos\kappa\cos p & -\cos\omega\sin\kappa + \cos\kappa\sin\omega\sin p & \sin\kappa\sin\omega + \cos\kappa\cos\omega\sin p \\ \cos p\sin\kappa & \cos\kappa\cos\omega + \sin\kappa\sin\omega\sin p & -\cos\kappa\sin\omega + \cos\omega\sin\kappa\sin p \\ -\sin p & \cos p\sin\omega & \cos\omega\cos p \end{pmatrix}$$

The camera internal parameters are: $x_p$ the size of a pixel in the x axis; s the ratio between a pixel in the x and y axes; $(p_x, p_y, p_z)$ the location in camera space of the principal point; (u,v) the location in image space of a an observed point (note: in our system the images have the (0,0) coordinate at the top left, so the y coordinate is reversed.

We now simplify this equation by substitution to obtain two equations, as follows. The first step is to divide the third

line into the first two in order to eliminate λ, and to simplify slightly.

$$a_1 + a_2 u + f(r[1,1]X + r[1,2]Y + r[1,3]Z + t[1])/U = 0$$
$$a3 + a4v + f(r[2,1]X + r[2,2]Y + r[2,3]Z + t[2])/U = 0$$
$$\text{Substitution:} a_1 \leftarrow -p_x, a_2 \leftarrow x_p, a_3 \leftarrow -p_y, a_4 \leftarrow -sx_p, U \leftarrow r_{20}X + r_{21}Y + r_{22}Z + t_z$$

We now eliminate $p_z$ (the focal length):

$$a'_1 + a'_2 u + (r_{00}X + r_{01}Y + r_{02}Z + t_x)/U = 0$$
$$a'_3 + a'_4 v + (r_{10}X + r_{11}Y + r_{12}Z + t_y)/U = 0$$
$$\text{Substitution:} a'_1 -> a_1/p_z, a'_2 -> a_2/p_z, a'_3 -> a_3/p_z, a'_4 -> a_4/p_z$$

Next we divide top and bottom by $t_z$:

$$a'_1 + a'_2 u + (r'_{00}X + r'_{01}Y + r'_{02}Z + t'_x)/U' = 0$$
$$a'_3 + a'_4 v + (r'_{10}X + r'_{11}Y + r'_{12}Z + t'_y)/U' = 0$$
$$\text{Substitution:} r'_{ij} \leftarrow r_{ij}/t_z, t'_i \leftarrow t_i/t_z, U' \leftarrow r'_{20}X + r'_{21}Y + r'_{22}Z + 1$$

Finally, we remove the $a'_i$, and then make a final substitution to get a linear form for the DLT:

$$l_1 X + l_2 Y + l_3 Z + l_4 + l_9 uX + l_{10} uY + l_{11} uZ = -u$$
$$l_5 X + l_6 Y + l_7 Z + l_8 + l_9 vX + l_{10} vY + l_{11} vZ = -v$$
$$\text{Substitution 1:} r''_{0j} \leftarrow (r'_{0j} + a'_1 r'_{2j})/a'_2, r''_{1j} \leftarrow (r'_{1j} + a'_3 r_{2j\prime})/a'_4,$$
$$t''_x \leftarrow (t'_x + a'_1)/a'_2, t''_y \leftarrow (t'_y + a'_3)/a'_4$$
$$\text{Substitution 2:} l_1 \leftarrow r''_{00}, l_2 \leftarrow r''_{01}, l_3 \leftarrow r''_{02}, l_4 \leftarrow t''_x, l_5 \leftarrow r''_{10}, l_6 \leftarrow r''_{11},$$
$$l_7 \leftarrow r''_{12}, l_8 \leftarrow t''_y, l_9 \leftarrow r'_{21}, l_{10} \leftarrow r'_{21}, l_{11} \leftarrow r'_{22}$$

We end up with two linear equations in eleven unknowns. A minimum of five points is required to provide a solution, and the points cannot be colinear. This point is discussed in

slightly more detail below. Given the $l_i$ it is possible to recover the basic camera parameters, as follows:

$$
\begin{aligned}
t_z &= \sqrt{(1/(l_8^2 + l_9^2 + l_{10}^2)} \\
p_x &= -x_p t_z^2 * (l_1 * l_9 + l_2 * l_{10} + l_3 * l_{11}) \\
p_y/s &= x_p t_3^2 * (l_5 * l_9 + l_6 * l_{10} + l_7 * l_{11}) \\
p_z &= x_p \sqrt{((t_z^2 * (l_1^2 + l_2^2 + l_3^2) - (p_x/x_p)^2)} \\
s &= \sqrt{(p_z^2/(x_p^2 t_z^2 * (l_5^2 + l_6^2 + l_7^2) - (p_y/s)^2))} \\
r_{12} &= t_z(s x_p l_7 + p_y l_{11})/p_z \\
t_x &= (p_x - x_p l_4) t_z/p_z \\
t_y &= (s x_p l_8 + p_y) t_z/p_z
\end{aligned}
$$

The angles can be recovered using the arcsin function using the following relations:

$$
\begin{aligned}
\sin p &= -x_p t_z((p_x l_{11}/x_p - l_3)/p_z) \\
\cos p &= \sqrt{(l_{11} * t_z)^2 + r_{12}^2} \\
r_{00} &= t_z(p_x l_9 - x_p l_1)/p_z \\
r_{01} &= t_z(p_x l_{10} - x_p l_2)/p_z \\
\cos \kappa &= r_{00}/\cos p \\
\sin \kappa &= r_{01}/\cos p \\
\sin \omega &= r_{12}/\cos p \\
\cos \omega &= t_z l_{11}/\cos p
\end{aligned}
$$

**ELLIPSE CORRECTION**

The image of a circle in a camera is an ellipse. The center of the ellipse in the image plane is not necessarily co-incident with the image of the circle's center, due to perspective distortion. This effect is shown, in exaggerated form for two dimensions ("2-D") in Figure 19. The camera has principal point P. Its focal plane is AC and the line AB is the cross section of an ellipse. $O_1$ is the center of the

projection of the line on the image plane, and $O_2$ is the projection of the center of the ellipse O. We now derive an expression for the difference between the observed center of the ellipse in an image, and the projected center of the circle. This correction is used by our system in its photogrammetric calculations.

Without loss of generality we assume that the ellipse is of unit radius, in the x-y plane, with center at (0,0,0). This situation can always be obtained by a suitable rigid body transformation of the camera and scaling of the focal length of the camera. This derivation explains the principle behind the specific calculations carried out in the system.

The ellipse is expressed parametrically in homogenous coordinates as $e_t = (\sin t, \cos t, 0, 1)^T$.

The camera has a rotation **R** and translation **T** w.r.t. the coordinate frame of the ellipse. We represent this rigid body transform with a 4 x 4 homogenous matrix **M**:

$$M = \begin{pmatrix} r_0 & r_1 & r_2 & t_x \\ r_3 & r_4 & r_5 & t_y \\ r_6 & r_7 & r_8 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In calculating the projection of the circle and its center onto the image plane, we are interested only in the distance between the two (in meters). The projection matrix therefore needs only consider the focal distance of the lens, and not the principal point. We also ignore the effects of lens distortion, which over the distances being considered have only a second-order effect. We represent to focal distance by

$f = r/p_z$, where $p_z$ is the distance from the projective center of the camera to the focal plane, and r is the radius of the circle. The projection matrix P is

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & f & 0 \end{pmatrix}$$

We can now derive a parametric expression for the ellipse and its center $c^p$ in the image plane. The center is $c^p = P\ M\ (0,0,0,1)^T$ and the ellipse is $e^p{}_t = P\ M\ e_t$. Determination of the center of the ellipse in the image is done by locating the two points of the ellipse tangent to lines parallel to the y-axis. The mid-point of the line connecting these two points is the center of gravity of the ellipse. These points are located by:

1. Differentiating the ellipse w.r.t $t$, which gives $(e_x, e_y, e_w)^T = de^p/dt$, and is algebraic in both sin$t$ and cos$t$
2. In order to avoid a transcendental equation, we make a substitution, $u \leftarrow \sin t$, $\checkmark(1-u^2) \leftarrow \cos t$
3. We solve the equation $e_x/e_w = 0$ for u, which is quadratic in u and gives (in general) two real solutions for the tangent points.

The ellipse equation in the plane in homogenous coordinates is:

$$e^p = (t_x + r_1 \cos t + r_0 \sin t, t_y + r_4 \cos t + r_3 \sin t, f t_z + f r_5 \cos t + f r_4 \sin t)^T$$

Differentiating the ellipse and performing the change of variables $u \leftarrow \sin t$, $\checkmark(1-u^2) \leftarrow \cos t$ gives:

$$\frac{de^p(x)}{dt} = \frac{r_0 r_5 - r_1 r_4 - r_4 t_x u + r_0 t_z u + r_5 t_x \sqrt{1-u^2} - r_1 t_z \sqrt{1-u^2}}{f(t_z + r_5 u + r_4 \sqrt{1-u^2}}$$

where $de^p(x)$ is the affine value of the x-coordinate of $de^p$.

To determine the zeros of this equation w.r.t u we need only consider the zeros of the top half of the fraction. If we make the substitution $a \leftarrow r_0 r_5 - r_1 r_4$, $b \leftarrow r_0 t_z - r_4 t_x$, $c \leftarrow r_5 t_x - r_1 t_z$ we have an equation of the form $a + bu + c\sqrt{(1-u^{2)}} = 0$ which has solution:

$$u = \frac{-2ab \pm 2\sqrt{a^2 b^2 - (a^2 - c^2)(b^2 + c^2)}}{2(b^2 + c^2)}$$

Given the values for u we can back-substitute in the equation for $e^p$ to determine the extremal points $p_0$ and $p_1$. The center of the ellipse is $(p_0 + p_1)/2$.

**THE PREFERRED EMBODIMENT OF THE LEVENBERG MARQUAND ALGORITHM**

Photogrammetry in our system requires the solution of non-linear least squares equations. We use a variant of the Levenberg Marquand algorithm. The non-linear least-squares problem is to find a vector of parameter values, **a**, to minimize the sum of squares of differences between a given vector, **y**, of observed values and a vector of fitted values $f(\mathbf{a}; \mathbf{x})$ where the values of **x** and the function $f$ is known.

When we are doing bundles adjustment the values $y_i$ are the observations of the image coordinates of known world points, the known parameters $\mathbf{x}_i$ are the positions of the known world

points, and the parameters $\mathbf{a}_i$ are the internal and external orientation parameters of the camera(s). We assume that the errors in observations are all normally distributed. We discuss how the variance of different measurements can be incorporated below.

We start with an estimate $\mathbf{a}_0$ of $\mathbf{a}$, whch will be iteratively refined. Initially we have the relationship:

$$\mathbf{y} = f(\mathbf{a}_0; \mathbf{x}) + \epsilon_0$$

We assume that $\mathbf{a} = (a_0, \ldots, a_m)$, and that $\mathbf{y} = (y_0, \ldots, y_n)^T$. We write the i-th equation as $y_i = f(\mathbf{a}; \mathbf{x}_i) + \epsilon_0(i)$. We wish to find an $\mathbf{a}$ that minimises the 2-norm of $\epsilon$. We assume that $f$ can be approximated locally by first derivatives. Let J be the Jacobian, with entries $J_{ij} = \partial f_i / \partial a_j$. A better approximation for $\mathbf{a}$ is then $\mathbf{a}_1 = a) + J\Delta$ where $\Delta$ is chosen to minimise the 2-norm of $\mathbf{y} - {}_f(\mathbf{a}_0; \mathbf{x}) - J\Delta$. the value of $\Delta$ can be determined by the method of normal equations, which evaluates the pseudo-inverse of J. We have:

$$\Delta = (J^T J) J^T \epsilon_0$$

When we have some idea of the errors associated with individual measurements, we can incorporate a weight matrix into this equation

$$\Delta = (J^T W J) J^T \epsilon_0$$

where $W_{ij} \propto 1/\sigma^2{}_{ij}$ and $\sigma^2{}_{ij}$ is the covariance between the ith and jth measuments. In practice this matrix is usually diagonal, representing independent measurements.

Once $\Delta$ is calculated we iterate until we are satisfied there is a solution. There are many choices of termination criterion (see Slama, referred to above). The advantage of this Newton-like method is that when the function $f$ is reasonably well behaved (and models reality), and the initial approximation $\mathbf{a}_0$ is close to the correct value we get quadratic convergence. The problem with the Newton-like approach is that when the function is not well approximated by the Jacobian the solution will not be found. The Levenberg-Marquand algorithm is a variation of the Newton method which improves stability by varying between the Newton step and a direct descent method. The Levenberg-Marquand algorithm uses the following formula to determine $\Delta$.

$$\Delta = (J^T W J + \lambda D) J^T \epsilon_0, D = \mathrm{diag}(J^T W J)$$

At the first iteration $\lambda$ is set to a relatively large number (say 1). If the first iteration reduces $\epsilon$ then $\lambda$ is reduced by some factor (say 5-10), otherwise $\lambda$ is increased by some factor and we keep trying until $\epsilon$ is reduced.

The Levenberg-Marquand method converges upon a stationary point (or subspace if the system is degenerate) which could be the global minimum we want, a local minimum, a saddle point, or 1 or more of the parameters could diverge to infinity.

The standard references are K Levenberg: "A Method for the Solution of certain non-linear problems in least squares" (Quarterly Applied Math., 2:164-168,1944) and D M Marquardt : ("Journal of the Society for Industrial and Applied Mathematics, 11:431-441, 1963). Robust public domain

implementations are available. The preferred implementations used in our system are the Linpack version as described by Moré (see above reference) and the implementation described by Dennis et al (see above reference) which is somewhat more robust.

**STEREO MATCHING**

The purpose of stereo matching is to construct a disparity map from a pair of digitised images, obtained simultaneously from a pair of cameras. By a disparity map we mean a two dimensional array which specifies for each pixel $p_l$ in the left image of the pair, the distance to a corresponding point $p_r$ in the right image (i.e. the stereo image disparity). By a point we mean a pair of real valued co-ordinates in the frame of reference in the right image. By a corresponding point we mean one that most probably corresponds to the point in the right image, at which the scene component $\pi_s$ imaged by $p_l$ in the left camera, will appear when imaged by the right camera.

The present technique uses five main data structures each of which is an image in the form of a two dimensional array of real numbers:

1. The left image L, derived from the left camera 3 in which the real numbers represent brightness values
2. The right image R derived from the right camera 4 in which the real numbers represent brightness values
3. The horizontal displacement image H. This specifies for each pixel (x,y) in the left image the displacement from x at which the corresponding point occurs in the right image.

4.The vertical displacement image V, with similar properties.

5. The confidence image C, which specifies the degree of confidence with which the disparity is held.

At any stage in the process the best available estimate of the disparity map is provided by H and V. Thus to pixel $L_{x,y}$ there corresponds a point $R_{x+(Hx,y),y+(Vx,y)}$. This will in general have real valued indices.

**FORMING THE SCALE PYRAMID**

The images L and R are available to the algorithm at multiple scales with each scale being the result of a decimation and filtering process as follows. An image pyramid is formed for each of L and R, with n+1 levels, labelled 0, 1, ..., n. If $P_i$ is an image in the pyramid at scale i and if $P_{i-1}$ is an image at the previous, larger, scale (which will be the next level down in the pyramid), by which we mean that if $\delta(P_i)$ is the length of the diagonal of the image at scale i then $\delta(P_i) = f\delta(P_{i-1})$ for some f in the range $0 < f < 1$, then

$$P_i = \text{scale}(G_{i+1}, 1/f) - G_i$$

where $G_i$ $(0 \leq i \leq n+1)$ is a Gaussian convolution defined by:

$G_0$ = convolve(I), where I is either L or R, at the largest scale

$G_{i+1}$ = scale(convolve($G_i$),f) for all other scales.

Convolve is a function returning a Gaussian convolution of an image; scale(I,f) is a function that scales an image I by the real number f. In the present described embodiment the scale

factor is 0.5, such that each next level in the pyramid is scaled by a factor of 0.5 relative to the level below it (in each linear dimension i.e. x and y dimensions).

If we define a convolution of an image I at point (x,y) by an n x n matrix K to be conv(I,x,y,K) where

$$conv(I,x,y,K) = \sum^{n-1}_{j=0} \sum^{n-1}_{i=0} I_{x-m+i,y-m+j} K_{i,j}$$

where m = (n-1)/2. For the Gaussian convolution above the kernel is given by $K_{i,j} = \omega(i)\omega(j)$ where

$$\omega(i) = \omega'(i) / [\sum^{m}_{j=-m} \omega'(j)]$$

and

$$\omega'(j) = (1/N) . (\sum^{N-1}_{u=0} g(j - 1/2 + u/(N-1); \sigma )$$

N is the number of sample steps used to approximate the value of the Gaussian distribution for any given point in the weight vector = $\omega'$ and

$$g(x,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} . e^{\frac{x^2}{2\sigma^2}}$$

In practice the scaling and convolution are applied at the same time using separable horizontal and vertical convolution kernels of size 5. Thus each image $P_i$ contains only information at a characteristic frequency, all higher and lower frequencies having been removed by the filtering process.

The present stereo matching technique uses a scale pyramid formed by a difference of Gaussian (DoG) convolutions as immediately above-described. Fig. 20 is a flow diagram

illustrating, in principle, the method used to obtain the DoG pyramid. Fig. 20 shows how each input image is used to obtain the corresponding image at each level in the pyramid.

A matching process is run on each successive scale starting with the coarsest (i.e. top-most level of the pyramid), as will be described. In moving to successively greater levels of detail (moving "down" the pyramid), the initial estimates of disparity for each level are provided from the results of matching the previous level. A flowchart for this process is provided in figure 21. The benefits of using a scale DoG pyramid are:

- It is easier to find the globally optimum disparity estimate, rather than local optima.
- The inter-scale relationship provides guarantees about the amount of search that must be done at a lower scale, given an estimate at a higher scale, and assuming that this estimate is accurate.
- The use of a DoG filter provides immunity from illumination changes across the two views of the scene.
- No user intervention to determine an initial set of matching points or regions is required.

Fig. 22 illustrates the key stages of the "matching" process carried out at any one scale of the scale pyramid and the sequence of these stages. It will be observed that the discrepancy and confidence maps (H,V,C) are circulated through four processing blocks 52,54,56,58 each cycle of the process. The L and R images for the appropriate level of the pyramid are input at the respective cylces of the process. The whole process can be seen as an iterative process aimed at arriving

at a good estimate of (H,V,C). It is run for five iterations in the preferred embodiment of the invention. The process will now be described in further detail.

**Initialisation**

The initialisation block 50 performs the following function:

**If** it is the start **then** set all pixels in H and V to 0.0 and set all pixels in C to 1.

**Otherwise** leave H, V, C with the values obtained from the last iteration of the smoother.

Starting with the pair of images from the top level of the pyramid (i.e. the "coarsest" images) we then move on to the "Warping" phase.

**Warping**

The "warping" block 52 takes as its input (R,H,V) and generates an output L', where $L'_{x,y} = R_{x+(Hx,y),y+(Vx,y)}$ For a perfect estimate of H and V, and for a scene with no occlusions we should have L = L'. (To the extent that they differ, either the estimated disparity is wrong or there are too many occluding edges).

Because L' is not necessarily on the integer image grid of L (i.e. L' may be at a fractional position between integer pixel positions in L), the value (which will be a real number representing brightness of the pixel) for L' is calculated using 4-point bilinear interpolation. Thus, the four pixels that enclose L' (in L) are used to estimate L'.

**Matching**

The correlation measure used by the matching block 54 is:

$$\mathrm{cor}_{l,r}(x,y) = \frac{\mathrm{cov}_{l,r}(x,y)}{\mathrm{var}_l(x,y)\mathrm{var}_r(x,y)}$$

with

$$\mathrm{cov}_{l,r}(x,y) = \sum_u \sum_v L_{(x+u,y+v)} R_{(x+u,y+v)} w_{(u,v)}, -2 \le u \le 2, 2- \le v \le 2$$

and

$$\mathrm{var}_i(x,y) = \sum_u \sum_v I_{(x+u,y+v)} I_{(x+u,y+v)} w_{(u,v)}$$

where $u$ and $v$ are integral pixel locations, and $\omega$ is a Gaussian kernel centered at (0,0).

The matching phase carried out by block 54 proceeds as follows:

1. For each pixel p=(x,y) in L
   (a) Let $\kappa_{x,y}(i,j)$ be the correlation between the neighborhood around $L_{x,y}$ and the neighborhood around $L'_{i,j}$.
   (b) Compute the horizontal correlations $\kappa_{x,y}(x-\delta,y)$, $\kappa_{x,y}(x,y)$, $\kappa_{x,y}(x+\delta,y)$.

$\delta$ is a value which is a (decimal) fraction of one pixel integer i.e. such that the location (x-$\delta$), for example, is an integral location along the x-axis, falling between integer pixel positions x and x-1. For example, the intial chosen value of $\delta$ might be 0.5.

   (c) Fit a parabolic curve through (i.e. in practice, fit a quadratic function to) these points and

(i) (a) if the curve has a negative second derivative and has a maximum $x_{max}$ in the range $x \pm 1.5\delta$, then (b) add $x-x_{max}$ to $H_{x,y}$ to give the new estimate of the horizontal disparity.

(ii) if the curve has a negative second derivative and a maximum outside this range set $x_{max} \leftarrow \pm 1.5\delta$ and proceed as in (i)(b) above.

(iii) if the curve has a positive second derivative, evaluate it at $(x\pm 1.5\delta, y)$ and set $x_{max}$ to whichever position yields the greatest predicted correlation and then proceed as in (i)(b) above

(d) proceed similarly in the vertical direction, updating $V_{x,y}$.

(e) set $\kappa_{max}$ to be the average of the maximums of the correlations in the horizontal and vertical directions.

(f) set $C_{x,y} \leftarrow 0.7\kappa_{max}+0.3C_{x,y}$

2. Set $\delta \leftarrow 0.5\delta$

**Smoothing**

The purpose of the smoothing or regularization phase (carried out by regulating block 56) is to adjust the disparity H, V and confidence C maps to arrive at a more accurate estimate of the disparity. The reason this is desirable is that the initial estimates derived from the matching phase are inevitable noisy. These estimates are adjusted, taking into account the strength of the correlation, and data from the original image.

1. New values of H, V, and C are obtained by applying a convolution to the neighborhoods surrounding each pixel in H, V, C such that:

$$H_{xy}^{new} = \mathbf{conv}(H, x, y, W(L, x, y, C))$$
$$V_{xy}^{new} = \mathbf{conv}(V, x, y, W(L, x, y, C))$$
$$C_{xy}^{new} = \mathbf{conv}(C, x, y, W(L, x, y, C))$$

where : W(I,a,b,P) is a weight construction function that computes a 3 x 3 convolution kernel on image I at co-ordinate a,b, using a probability array P. As shown above, in the present case the confidence map C is chosen as the probability array P.

**conv**(I,a,b,K) evaluates a convolution on image I at position a,b using kernel K.

The weight construction function currently used produces the matrix $\psi V$ where

$$\tilde{V} = \begin{pmatrix} 0 & P_{a,b+1}|I_{a,b+1} - I_{a,b}| & 0 \\ P_{a-1,b}|I_{a-1,b} - I_{a,b}| & P_{a,b} & P_{a+1,b}|I_{a+1,b} - I_{a,b}| \\ 0 & P_{a,b-1}|I_{a,b-1} - I_{a,b}| & 0 \end{pmatrix}$$

and

$$\psi = \left(\frac{1}{\sum_i \sum_j V_{ij}}\right)$$

2. Step 1 is repeated 20 times

Once the initialise 50, warping 54, matching 54, and smoothing (regularising) 56 phases have been carried out as above described, the system moves on to the carry out all of these

phases for the pair of images in the next level down (moving from coarse to fine) in the image pyramid. The complete stereo matching process is iterated through all the levels of the pyramid, as illustrated by Fig.21. (There are five iterations in the present described embodiment.)

It will be appreciated that before carrying out the warping stage at the next level down in the pyramid, the values of H and V (the horizontal and vertical disparities) must be appropriately scaled. (In the present embodiment H and V mst therefore be multiplied by the factor 2 when moving from one pyramid level down to the next level. This is because the scaling factor f used to calculate the DoG images is 0.5 in our embodiment, as mentioned above. It will though be appreciated that other scaling factors could be used in different possible embodiments.)

By way of illustration, Fig.23 shows in flow chart form the general scheme of the above-described stereo matching method, and the sequence of steps used to iterate through all levels in the pyramid.

Using the above-described technique, together with textured illumination of the object scene, we have been able to "match" the left and right images to an accuracy of 0.15 pixels over approximately 90% of the image.

**ALTERNATIVE METHOD USING BOTH LEFT-RIGHT AND RIGHT-LEFT DISPARITIES**

The accuracy of the reconstructed image can be increased by calculating both left to right and right to left disparities. This allows occluded areas to be detected. (Occluded areas

are those which are visible to only one of the pair of cameras 3.4). Since the disparities calculated for occluded areas are almost certainly erroneous, the confidence map can be adjusted to take this into account.

To detect occlusions the warping (block 52) and matching (block 54) are done twice, everything left (l) is swapped with that of right (r) the second time. After this we get ($H_l$, $V_l$, $C_l$) and ($H_r$, $V_r$, $C_r$). Subsequently occlusions are detected and incorporated and used to modify $C_l$ and $C_r$. Smoothing (block 56) is applied to both ($H_l$, $V_l$, $C_l$) and ($H_r$, $V_r$, $C_r$).

**OCCLUSION DETECTION**

The above-mentioned detection of occlusion is done as follows:

Consider pixel $L(_{x,y})$ of the left image. From ($H_l$, V+l) we get the corresponding position of the pixel in the right image, $R(_{x',y'})$. From ($H_r$, $V_r$) we get the corresponding position of the pixel in the left image, $L(_{x'',y''})$. If the Euclidean distance of $L(_{x,y})$ and $L(_{x'',y''})$ is greater than some threshold [which is preferably equal to one (1)], the pixel $L(_{x,y})$ of the left image is classified as an occluded pixel. The pixel value of $C_l$ is set 0 to indicate that this is an occlusion, and thus that we have no confidence in our disparity measurement. (In our system the minimum value of confidence C is 0.04). This is done for every pixel of $C_l$ and $C_r$.

**BUILDING A 3-D MODEL**

Once the left and right cameras are calibrated (using the afore-described calibration technique) (i.e. their internal and external orientation is known), and the afore-described (stereo) matching method has determined the disparity map

between the two images, it is relatively straightforward to then build a dense 3-D world model. Recall from above the camera parameters used by our system for a camera c:

$(r_x, r_y, r_z)$ The Eulerian rotation angles for the camera w.r.t. the default world co-ordinate frame. The (3 x 3) rotation matrix corresponding to these angles (in the order x-axis, y-axis then z-axis is $\mathbf{R}_c$.

$(t_x, t_y, t_z)$ The translation for the camera w.r.t. the default world co-ordinate frame, let $t_c = (t_x, t_y, t_z)^T$

$(u_c, v_c, f)$ The principal point of the camera w.r.t. the camera frame. $u_c$ and $v_c$ are in pixel co-ordinates. The translation from camera space to pixel coordinates involves:

$x_p$ The size of an image pixel along the x-axis.

s The ratio between an x-pixel and y-pixel.

$(k_1, k_2, k_3, k_4, k_5)$ Distortion parameters for the camera.

Given this information about both the left and right cameras, and a pixel location $(u_l, v_l)$ in the left camera $c_l$, we can calculate the world position of this point as follows.

Let the disparity at $(u_l, v_l)$ be $(d_x, d_y)$. The coordinate of the corresponding point in the right camera $c_r$, is then $(u_r, v_r) = (u_l, v_l) + (d_0, d_y)$.

For each camera 3,4 the vector from the principal point to the image point (u,v) in pixel co-ordinates is $\mathbf{p}= ((u-u_c)x_p, (v-v_c)x_p s, -f)^T$. Let these vectors for $c_l$ and $c_r$ be $\mathbf{V_c}$ and $\mathbf{V_r}$. Let $\omega_l = \mathbf{R_l V_l}$ and $\omega_r = \mathbf{R_r V_r}$. $\mathbf{c} = \mathbf{t_l} - \mathbf{t_r}$. Let

$$t = \begin{pmatrix} \vec{w_r}\cdot\vec{w_r} & -\vec{w_l}\cdot\vec{w_r} \\ -\vec{w_l}\cdot\vec{w_r} & \vec{w_l}\cdot\vec{w_l} \end{pmatrix}^{-1} \begin{pmatrix} \vec{w_r}\cdot\vec{c} \\ -\vec{w_l}\cdot\vec{c} \end{pmatrix}$$

Then the point in space $\mathbf{p} = t_{cl} + t_x\omega_l$ .

This formula is used for each pixel in the left image L, and produces an output image, in one-to-one correspondence with the left camera image L, which contains a 3-D world position at each pixel.

Given one or more dense 3-D models, represented as an image in memory, with each pixel containing the distance from the principal point to the object surface, and given the internal and external orientation parameters of the camera for each model, the present invention includes a method for generating triangular polygon meshes from these models for the purposes of display and transmission. In addition a novel method of merging render images associated with each mode has been developed.

**Method for Building Polygon Meshes from Dense 3-D images**

In order to combine multiple dense models an intermediate structure is used, namely a 3-D voxel image. The preferred implementation is a variant of the that processed by Curless and Levoy (referenced above).

This image is an array of points $v_i,y_j,z_k, = (x_i,y_j,z_k)^T$, $0 \leq i < n_x$, $0 \leq i < n_y$, $0 \leq k < n_z$. This array is arranged uniformly in space with separation s, so that $v_{i+1,j+1,k+1} - v_{i,j,k} = (s,s,s)^T$.

Each point is categorized as UNSEEN, EMPTY or BOUNDARY. An UNSEEN point is one which lies between the principal point of a camera, and the model surface seen from that camera, with a distance greater than a threshold $\tau$. A BOUNDARY} point is one which has a distance from the principal point within tolerance $\tau$ of that seen in the model. Note that boundary points contain a signed distance. Points start with label UNSEEN.

Before describing the method, we first introduce some terms. Let the model image be $I = I(_{x,y})$, $(x,y) \in (0,0)-(n,m)$. Let the corresponding confidence map C derived from the initial match (see above) be the image $W = W(_{x,y})$, $(x,y) \in (0,0)-(n,m)$. Let the projection matrix that maps points in space to image co-ordinates in the image from camera M be $\mathbf{P}_m$. Let the axis vectors in space be $\mathbf{i},\mathbf{j},\mathbf{k}$, where $\mathbf{i} = (1,0,0)^T$ for example. Let the scale of camera c be $s_c = 1/\max\{||\mathbf{P}_c\mathbf{i}||_2, ||\mathbf{P}_c\mathbf{j}||_2, ||\mathbf{P}_c\mathbf{k}||_2\}$. Given an image point $\mathbf{u} = (u_x,u_y)$ in I with real co-ordinates, the value of $I_u$ if it is in the image, is computed by bilinear interpolation from the 4 neighboring pixels. Let the 2-distance of any point $\mathbf{v}$ in space from the principal point of camera c be $d_c(\mathbf{v})$. Let the cosine of the ray from the principal point of c to the model surface at image point $I_u$ be $Co_I(\mathbf{u})$

The method of adding a dense 3-D model, represented by a (float coded) image array $I_u$ and camera c, to an image volume V with distance threshold $\tau$ and confidence threshold $\omega_t$ is as follows:

1. If no models have yet been added to V, initialize all points in V to UNSEEN. We assume that the weight and distance of this point are both initialized at 0.
2. If $s_c < 1$ scale I and W by $s_c$, and scale the focal length of c by $s_c$. This ensures that the sampling of the image by V will not alias.

3. Process the confidence image W, by:
   (a) Setting all pixels below $\omega_t$ to 0.
   (b) Blob filtering W to remove all blobs of non-zero pixels, whose areas relative to that of the whole image falls below 3%.
   (c) Reduce the weight of non-zero pixels, if they are near the edge of a blob.
4. For each point $\mathbf{v} \in \mathbf{V}$ compute $\mathbf{u} = \mathbf{P}_c\mathbf{v}$. If $\mathbf{u}$ is in I then let $d_i = I_u$, $\omega_i = W_u$, $d_v$ be the distance from $\mathbf{v}$ to the principal point of c, and $Co_i = Co_I(\mathbf{u})$. Let $d = d_i - d_v$.

   (a) If $Co_i d > \tau$ then set $\mathbf{v}$ to EMPTY.
   (b) else if $Co_i d < \tau$ ignore $\mathbf{v}$
   (c) else if $\omega_i > 0$ let the current distance of $\mathbf{v}$ be $d_0$ and let the current weight be $\omega_0$, set the distance of $\mathbf{v}$ to $(d\omega_i + d_0\omega_0)/(\omega_i + \omega_0)$ and set the weight of $\mathbf{v}$ to $\omega_i + \omega_0$.

Once this image has been computed it is possible to triangulate this mesh using a suitable known isosurface

extraction method (e.g. as proposed by Lorensen and Cline in SIGGRAPH '87 Conference Proceedings (Analeim CA), pages 163-170; or Howie & Blake in Computer Graphics Forum, 13(3) : C/65-C/74, October 1994); or Livnat et al, IEEE Transactions on Visualisation and Computer Graphics, 2(1) : 73-84, March 1996).

The efficiency of the process of constructing the voxel image can be improved by two algorithmic methods.

**1) Run-length Encoding**

Since it is not necessary to make distinctions between any two points with labels UNSEEN or EMPTY, it is possible to run-length encode the voxel image. This is done by storing the x-ordinate as a variable run-length encoded array. Since the principal mode of access to the voxel image is iterative, this can be done without a performance penalty.

The amount of space required by the run-length encoded image is a function of the complexity of the surface. Typically each non-empty x-row will intersect the surface twice. Assuming the depth of the boundary cells is fixed, the space requirement is quadratic in the linear dimension of the volume, rather than cubic.

The memory organisation of the run-length encoded volume is amenable to multi-processing, whereby multiple range images can be simultaneously added to the volume. The most convenient way to do this is for each range image to be added using a random permutation of the x-coded arrays. This minimises possible memory contention.

**2) Pyramid coding**

When the number of voxels is very large most of the processing undertaken is redundant examination on non-BOUNDARY cells. Use of a voxel pyramid can significantly reduce processing time by focusing on BOUNDARY cells, the number of which grows only quadratically with linear volume dimension.

Each level of the volume pyramid is constructed and processed as described above, with some minor variations to be described below. The bottom level of this pyramid is described in section above entitled "Method for Building Polygon Meshes from Dense 3-D images". The other levels are scaled by a factor of 0.5 relative to the level below them. Let there be n levels to the pyramid, with the bottom level numbered n-1 and the top level 0. Using the terminology of the afore-mentioned above section, the voxel size at level I is $s_i$ and the distance tolerance is $\tau_i$.

In order to avoid aliasing the model images are, if necessary, scaled using a box filter to ensure that the projection of the vectors $s_i\mathbf{i}$, $s_i\mathbf{j}$, $s_i\mathbf{z}$ in the image are less than one pixel in size.

In order to add a new model $I_u$ and camera c, to an n-level image pyramid volume V with distance threshold $\tau_n$-1 and confidence threshold $\omega_t$ to the pyramid the following extension to the algorithm used in the section entitled "Method for Building Polygon Meshes..." above, is used:

1. The value of $s_n$-1 is chosen. The other values of $s_i$ are set at $s_i = s_{i+1}/2$, $I \in \{0, n-2\}$.

2. The value of $\tau_n$-1 is chosen. The other values of $\tau_n$ are set as $\tau_I = \max\{\tau_n+1/2, 2s_i\}$, $I \in \{0,n-2\}$.

3. The top level (level 0) of the pyramid is processed as described in the afore-mentioned above section entitled "Method for Building Polygon Meshes...".

4. For subsequent levels of the pyramid, computation is only done for those voxels with the label BOUNDARY in the level above. When the voxel at level I is processed with a distance greater than $\tau_i$, it is set to EMPTY. When the voxel is processed with distance value less than $-\tau_I$ it is set to UNSEEN and then processed in the same manner as the previous algorithm.

the specific representation used for voxel volumes, and the manner of processing them provides both a very significant speed-up over single-level computation, as well as improved memory performance.

**APPLYING SEAMLESS RENDER**

The render images captured from the center cameras of the pods, can be used to provide an image texture to a polygon mesh M, constructed as described above. The present invention contains a novel method for merging the different images to provide seamless texturing of the polygon mesh. Before describing the merging algorithm in detail, we describe the method by which render images, captured from the central (or left) cameras of the 3-D camera apparatus can be mapped onto polygon meshes.

Let the image to be mapped be $I_u$, as above. Let the camera

associated with the image be c. Let the vertices of the polygon mesh be $V = \mathbf{v}_i$, $\mathbf{i} \in I$. Let the normal of each vertex be $\mathbf{v}^n_i$ . If the projection matrix of camera c is $\mathbf{P}_c$ as above, and the vector from the principal point of c to $v_i$ is $\omega_i$ then we associate a texture coordinate $\mathbf{u} = \mathbf{P}_c\mathbf{v}_i$ with $\mathbf{v}_i$ ifthe vertex is visible from camera c.

Informally, a vertex is visible if (a) the surface at that point is facing the camera, and (b) no other part of the sirface is between the camera and the vertex. The test for (a) is whether $\mathbf{v}^n_i.\mathbf{w}_i < 0$. The test for (b) is performed by a standard hidden surface computation on the whole mesh.

With this preliminary we turn to the seamless merging of texture images. The images be $I^k$, $k \in \{1,\ldots,n\}$, the confidence images be $W^k$, $k \in \{1,\ldots,n\}$, the cameras be $c^k$, $k \in \{1,\ldots,n\}$, and the vertices be $V = \mathbf{v}_i$, $\mathbf{i} \in I$ as above.

The goal of the merging method is to merge seamlessly, and with as little blurring of the image as possible. Area-based merging techniques are likely to introduce blurring of the images due to misregistration and to differences in illumination between images. In order to obtain a seamless merge of the textures with minimal blurring we use a boundary-based approach. Each triangle in the mesh M is projected by one or more of the texture images. We determine which image projects to which triangles by optimising a combination of the confidence, or weight, associated with each triangle, and the size of connected patches to which an image projects. Before presenting this algorithm we define the neighbors of a triangle t as the set $N_t$. A neighbor is a triangle that shares

an edge. The algorithm for determination of which images project to which triangles is as follows:

1. For each traingle t in the mesh associate the image $I^k$ for which t has the highest confidence. The confidence $C_t$ of triangle t is defined as the average of the weights of its vertices with respect to the image. If any weight is missing, or the point does not project to the image the triangle's confidence is 0. We define a function m, where m(t) is the image $I^k$ with the maximum value of $C_t$.

2. Generate a partition P of the triangles of M where each element p of P is maximal in the sense that there do not exist t, t'$\in$ M such that t $\in$ p and t' $\notin$ p with t $\in N_{t'}$.

3. We say that a triangle t $\in$ M is consistent with an image I and camera c if $C_t$ is non-zero. We extend this definition to an element p of partition P in the natural way. We reduce the cardinality of P by repeating the following steps:

(a) Sort P by cardinality

(b) For each element p, smallest first, if p is consistent with an image I and there is a triangle t $\in$ p neighboring a triangle in a different element p' $\in$ P and the cardinality of p is not greater than that of p' then assign each element of p to a partition p' and reset m(t) for each t $\in$ p.

On termination of this algorith each triangle t is associated with an image $I^k$ via the partition P. Consider the vertices $v_i$, i $\in$ I that belongs to triangles in more than one partition of P. These are vertices on the boundary between two partitions. These vertices will be used to define the merge between

textures. We extend the set I with additional elements correspomding to new vertices. These vertices are positioned between vertices $v_i$, $i \in I$ connected by edges in M. The number of additional vertices to be added is user-specified. The rational for this addition is to ensure that the boundaries when projected onto the images are dense and will provide a smooth merge.

For each $i \in I$ let the subset of images in which $\mathbf{v}_i$ has associated texture co-ordinate $\mathbf{u}$ be $S_i \subseteq \{1,\ldots,n\}$, and let the texture co-ordinates be $\mathbf{u}^s_i$, $\mathbf{s} \in \mathbf{S}$, and the weights be $\omega^s_i$, $\mathbf{s} \in \mathbf{S}_i$.

These points are the projection of the point $\mathbf{v}_i$ in each image. In general the pixel intensities of these image points will be different from each other. In order to merge the textures, we define a new common value at this image point in each of the image $I^s$, $s \in S_i$. This value is weighted by the corresponding pixel weight in each image, and is

$$\frac{\rho_i = \sum_{S_i} w_i^s \mathbf{u}_i^s}{\sum_{S_i} w_i^s}$$

We also define $\delta^s_i = \rho_i - \mathbf{u}^s_i$, which is the difference between the original value at a pixel in the image, and its new value.

For each image $I^k$, $k \in \{1,\ldots,n\}$, let the set of points in V where $\rho_i$ is defined over more than one image, be $\Lambda^k \subseteq I$. Let the rectangular domain over which $I^k$ is defined be $\Omega_k \subset R^2$. The method of texture integration is based on determining a function $\varnothing_k$: $R \rightarrow R$, which is as "smooth" as possible subject

to the pointwise constraint: $\varnothing_k(\mathbf{u}_i) = \rho_i$, $\mathbf{I} \in \Omega_k$. Given this function, each image $I^k$ can be "warped" by adding the value of $\varnothing_k$ evaluated at a pixel to that pixel.

This is a version of the problem of data interpolation from scattered points. The preferred embodiment of the 3-D camera apparatus uses the method described by S Lee et al : "Scattered Data Interpolation with Multi level B-Splines", IEEE Transactions on Visualisation and Computer Graphics, 3(3) : 228-244, 1997.

CLAIMS

1. A method of measuring stereo image disparity for use in a 3-D modelling system, the method comprising the steps of:

(a) producing a first camera output image of an object scene;

(b) producing a second camera output image of said object scene;

(c) digitising each of said first and second camera output images and storing them in storage means;

(d) processing said first and second digitised camera output images so as to produce an image pyramid comprising a plurality of successively produced pairs of filtered images, each said pair of filtered images providing one level in the pyramid, each successive pair of filtered images being scaled relative to the pair of filtered images in the previous pyramid level by a predetermined amount and having coarser resolution than the pair of filtered images in said previous pyramid level, and storing these filtered images;

(e) calculating an initial disparity map for the coarsest pair of filtered images in the pyramid by matching one image of said pair of coarsest filtered images with the other image of said coarsest pair of filtered images;

(f) using said initial disparity map to carry out a warping operation on one image of the next-coarsest pair of filtered images in the pyramid, said warping operation producing a shifted version of said one image; and

(g) matching said shifted version of said one image of the next-coarsest pair of images with the other image of said next-coarsest pair of images so as to obtain a respective disparity map for said other image and said shifted image, which respective disparity map is combined with said initial

disparity map so as to obtain a new, updated, disparity map for said next-coarsest pair of images; and

(h) repeating steps (f) and (g) for the pair of filtered images in each subsequent pyramid level, at each level using the new, updated disparity map obtained for the previous level as said initial disparity map for carrying out the warping process in step (f), so as to arrive at a final disparity map for the least coarse pair of images in the pyramid.

2. A method according to claim 1, wherein said processing step (d) for image pyramid generation comprises operating on said first and second digitised camera output images with a scaling and convolution function.

3. A method according to claim 2, wherein the plurality of pairs of filtered images produced by said scaling and convolution function are Difference of Gaussian (DoG) images.

4. A method according to any preceding claim, wherein the first pair of filtered images produced in step (d) are of the same scale as the digitised first and second camera output images, and each subsequent pyramid level contains images which are scaled by a factor of f, where $0 < f < 1$, relative to the previous level.

5. A method according to any preceding claim, wherein there are at least five levels in the image pyramid.

6. A method according to any preceding claim, wherein the method includes repeating steps (f) and (g) at least once,

at at least one of said pyramid levels, using the latest new, updated disparity map for each warping operation.

7. A method according to any preceding claim, further including constructing a confidence map during each iteration of steps (f) and (g) of the method, the contents of said confidence map constructed at any one level of the pyramid representing the confidence with which the contents of the disparity map at said one level are held.

8.A method according to claim 7, further including the step of carrying out a smoothing operation on the disparity and confidence maps produced at each level in the image pyramid, prior to using these smoothed maps in the calculation of the new, updated disparity maps and confidence maps at the next level.

9. A method according to claim 8, wherein said smoothing operation comprises convolving each said map with a predetermined weight construction function W(I,a,b,P).

10. A method according to claim 9, wherein said weight construction function W(I,a,b,P) is dependent upon original image intensity values, and confidence values, associated with each pixel.

11. A method according to claim 9, wherein said weight construction function W(I,a,b,P) computes a convolution kernel on a data array representing an image I, at a pixel p(a,b), using a probability array P, which is the confidence map C.

12. A method according to any preceding claim, wherein in steps (e) and (g) said matching process by means of which one image is matched with another is carried out in each case by:

(a) calculating horizontal correlation values for the correlation between a neighbourhood around each pixel p(x,y) in one image and neighbourhoods around at least three horizontally co-linear points in a spatially corresponding area of the second image, and fitting a parabolic curve to said horizontal correlation values and analysing said curve in order to estimate a horizontal disparity value for the said pixel p(x,y); and

(b) calculating vertical correlation values for the correlation between a neighbourhood around each pixel p(x,y) in one image and neighbourhoods around at least three vertically co-linear points in a spatially corresponding area of the second image, and fitting a parabolic curve to said vertical correlation values and analysing said curve in order to estimate a vertical disparity value for the said pixel p(x,y).

13. A method according to claim 12, wherein data values of the image for fractional points located between pixels are obtained by using bilinear interpolation.

14. A method according to any preceding claim, wherein each of said first and second output images comprises an array of pixels, and the final disparity map comprises the calculated disparity for each pixel in the first image relative to the second image, and the method further includes repeating steps (e) to (h) of the method to calculate a final disparity map for each pixel in the second image relative to the first image, and then comparing the two final disparity maps so as

to detect areas of the object scene which are occluded in one of said first and second output images.

15. A method according to any preceding claim, further including illuminating the object scene with a textured pattern.

16. a method according to claim 15, wherein the textured pattern comprises a digitally generated fractal random pattern of dots of different levels of transparency.

17. A 3-D image modelling system comprising:
first camera imaging means (3) for producing a first camera output image of an object scene;
second camera imaging means (4) for producing a second camera output image of said object scene;
digitising means (9) for digitising each of said first and second camera output images;
storage means (17,18) for storing said digitised first and second camera output images; and
image processing means (11) programmed to:
(a) process said first and second camera output images so as to produce an image pyramid of pairs of filtered, preferably Difference of Gaussian (DoG), images from said first and second digitised camera output images, each successive level of the pyramid providing smaller images having coarser resolution, and said storage means being capable of also storing the pairs of filtered images so produced;
(b) process the coarsest pair of filtered images in the pyramid so as to: calculate an initial disparity map for said coarsest pair of filtered images; use said initial disparity map to carry out a warping operation (52) on one said next-

coarsest pair of filtered images, said warping operation producing a shifted version of said one of said next-coarsest pair of filtered images; matching (54) said shifted version of said one of said next-coarsest pair of filtered images with the other of said next-coarsest pair of filtered images to obtain a respective disparity map for said other image and said shifted image, which disparity map is combined with said initial disparity map to obtain a new, updated, disparity map for said next-coarsest pair of filtered images;

(c) iterating said warping and matching processes for the pair of images at each subsequent level of the scale pyramid, at each level using the new, updated disparity map from the previous iteration as the "initial" disparity map for carrying out the warping step of the next iteration for the next level, prior to calculating the new, updated, disparity map at this next level, so as to obtain a final disparity map for the least coarse pair of filtered images in the image pyramid; and

(d) operating on said first and second digitised camera output images using said final disparity map, in a 3-D model construction process, in order to generate a three-dimensional model from said first and second camera output images.

18. A 3-D modelling system according to claim 17, further including projector means (13) for projecting a textured pattern onto the object scene.

19. A computer program product comprising:

a computer usable medium having computer readable code means embodied in said medium for carrying out a method of measuring stereo image disparity in a 3-D image modelling system, said computer program product having computer readable code means for:

processing data corresponding to a pair of first and second digitised camera output images of an object scene so as to produce filtered data corresponding to a plurality of successively produces pairs of filtered images, each pair of filtered images providing one level in the pyramid, each pair of filtered images being scaled relative to the pair of filtered images in the previous level by a predetermined amount and having coarser resolution than the pair of images in said previous level;

calculating an initial disparity map for the coarsest pair of filtered images by matching filtered data of one image of said coarsest pair of filtered images with the filtered data of the other image of said coarsest pair of filtered images;

using said initial disparity map to carry out a warping operation on the data of one image of the next-coarsest pair of filtered images in the pyramid, said warping operation producing a shifted version of said one image; and

matching said shifted version of said one of said next-coarsest pair of images with the other of said next-coarsest pair of images so as to obtain a respective disparity map for said other image and said shifted image, which disparity map is combined with said initial disparity map so as to obtain a new, updated, disparity map for said next-coarsest pair of filtered images; and

iterating said warping and matching processes for the pair of images at each subsequent level of the scale pyramid, at each level using the new, updated disparity map from the previous iteration as the "initial" disparity map for carrying out the warping step of the next iteration for the next level, prior to calculating the new, updated, disparity map at this next level, so as to obtain a final disparity map for the least coarse pair of filtered images in the image pyramid.

20. A computer program product according to claim 19, further including computer readable code means for:

operating on said first and second digitised camera output image data using said final disparity map, in a 3-D model construction process, in order to generate data corresponding to a three-dimensional image model from said first and second digitised camera output image data.

21. A method of calibrating cameras for use in a 3-D modelling system so as to determine external orientation parameters of the cameras relative to a fixed reference frame, and determine internal orientation parameters of the cameras, the method comprising the steps of:

(a) providing at least one calibration object having a multiplicity of circular targets marked thereon, wherein said targets lie in a plurality of planes in three dimensional space and are arranged such that they can be individually identified automatically in a camera image of said at least one calibration object showing at least a predetermined number of said circular targets not all of which lie in the same plane;

(b) storing in a memory means of the modelling system the relative spatial locations of the centres of each of said target circles on said at least one calibration object;

(c) capturing a plurality of images of said at least one calibration object with each of a pair of first and second cameras of the modelling system, wherein at least some points, on said at least one calibration object, imaged by one of said cameras are also imaged by the other of said cameras;

(d) analysing said captured images so as to: locate each said circular target on said at least one calibration object which

is visible in each said captured image and determine the centre of each such located circular target, preferably with an accuracy of at least 0.05 pixel, most preferably with up to 0.01 pixel accuracy; and identify each located circular target as a known target on said at least one calibration object;

(e) calculating initial estimates of the internal and external orientation parameters of the cameras using the positions determined for the centres of the identified circular targets.

22. A method according to claim 21, wherein step (e) is carried out using a Direct Linear Transform (DLT) technique.

23. A method according to claim 21 or claim 22, including the further step of:

(f) refining the initial estimates of the internal and external orientation parameters of the cameras using a least squares estimation procedure.

24. A method according to claim 23, wherein step (f) is carried out by applying a modified version of the co-linearity constraint, in a the form of an iterative non-linear least squares method, to the initial estimates of the internal and external orientation parameters of the cameras, to calculate a more accurate model of the internal and external orientation parameters of the cameras.

25. A method according to claim 24, where modelling of perspective distortion is incorporated in said iterative non-linear least squares method used to calculate a more accurate model of the internal and external orientation parameters of the cameras.

26. A 3-D modelling method incorporating the method of calibrating cameras according to claim 21, and the method for measuring stereo image disparity according to claim 1, and wherein the estimated internal and external parameters of the first and second cameras, and the final calculated disparity map for the first and second camera output images, are used to construct a 3-D model of the object scene.

27. A 3-D modelling method according to claim 26, wherein the 3-D model is in the form of a polygon mesh.

28. A 3-D modelling system according to claim 17 or claim 18, wherein the image processing means is further programmed to carry out steps (d) and (e) in the method of claim 21, and wherein the system further includes at least one said calibration object and storage means for storing constructed 3-D models.

29. A 3-D modelling system according to claim 17 or claim 18, wherein a plurality of pairs of left and right cameras are used, in order to allow simultaneous capture of multiple pairs of images of the object scene.

30. A 3-D modelling method according to claim 26 or claim 27, wherein multiple pairs of images of the object scene are captured and each pair of images is used to produce a 3-D model of the object scene, and the plurality of said 3-D models thus produced are combined together in a predetermined manner to produce a single output 3-D model.

31. A 3-D modelling method according to claim 30, wherein the plurality of 3-D models are combined in an intermediate 3-D voxel image which is then triangulated using an isosurface extraction method, in order to form the single output 3-D model which is in the form of a polygon mesh.

32. A method according to claim 31, further including integrating at least one render image onto said polygon mesh so as to provide texturing of the polygon mesh.

33. A method according to claim 32, wherein said at least one render image is integrated onto the polygon mesh by using a boundary-based merging technique.

34. A 3-D modelling system according to any of claims 17,18 and 28, wherein the system incorporates a camera pod comprising three cameras, namely said first and second camera imaging means for capturing left and right object scene images, and a third camera imaging means for capturing at least one image for providing visual render information.

35. A system according to claim 34, wherein the system incorporates a plurality of such camera pods.

36. A modelling system according to claim 28 or claim 29, wherein the or each said calibration object is provided with at least one optically readable bar code pattern which is unique to that calibration object, and the image processing means is programmed to locate said at least one bar code pattern and to read and identify said bar code pattern as one of a pre-programmed selection of bar code patterns stored in a memory means of the system, each said stored bar code pattern

being associated with a similarly stored set of data corresponding to a respective said calibration object.

37. A system according to claim 36, wherein the or each said calibration object is additionally provided with bar code location means in the form of a relatively simple locating pattern which the image processing means is configured to locate and, from the location of said locating pattern, identify that portion of the image containing said at least one bar code pattern, prior to reading said bar code pattern.

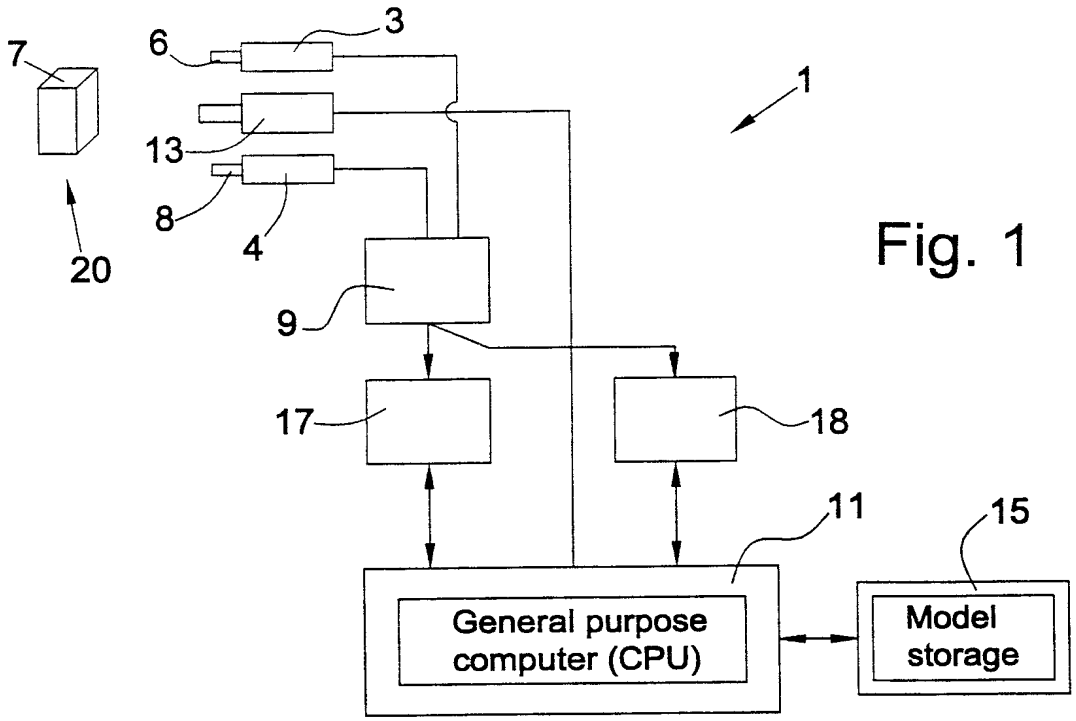
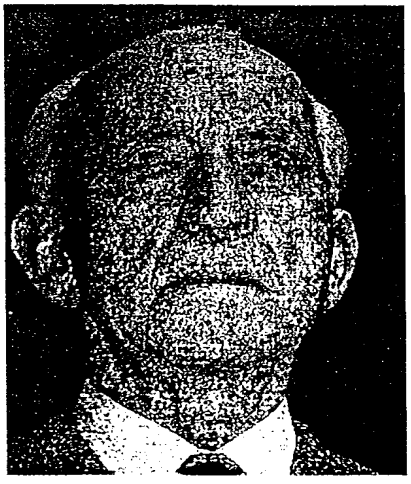
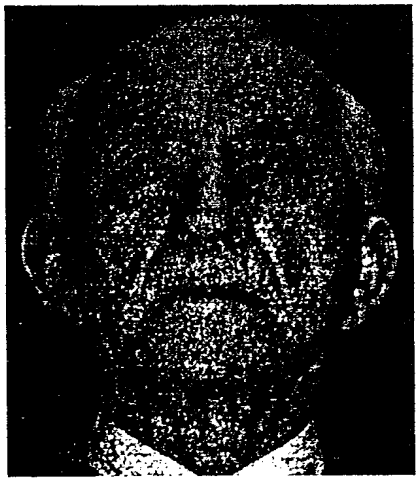



Fig. 1



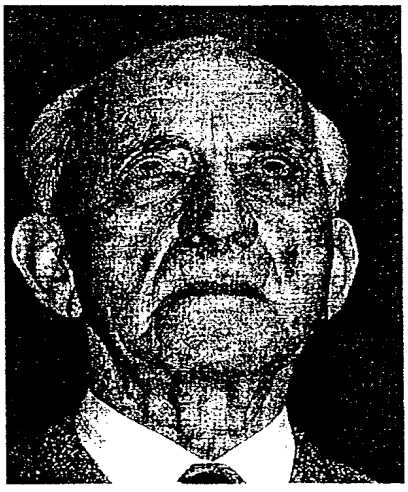
Left texture

Fig. 2(a)



Right texture

Fig. 2(b)



Left render

Fig. 2(c)

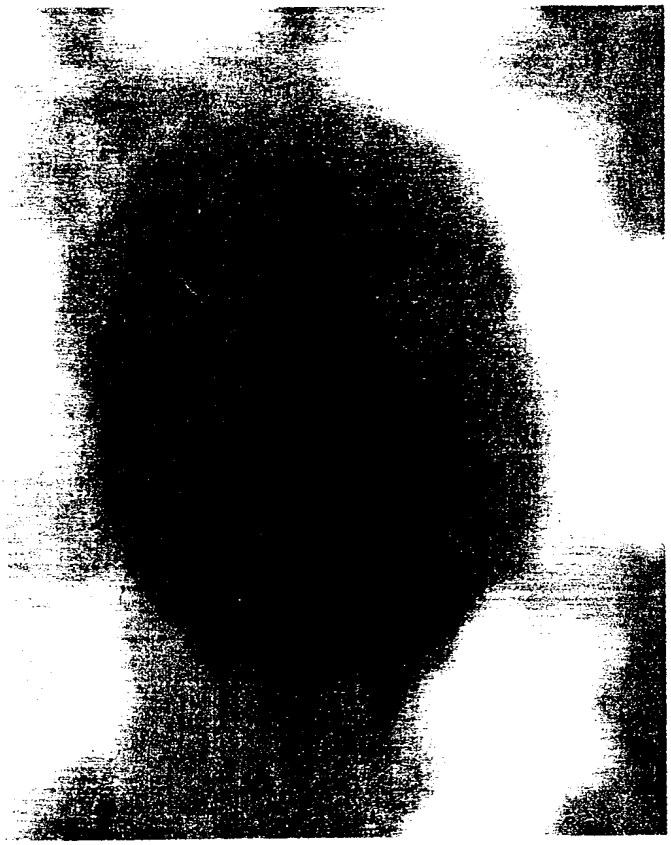


Fig. 3

Fig. 4

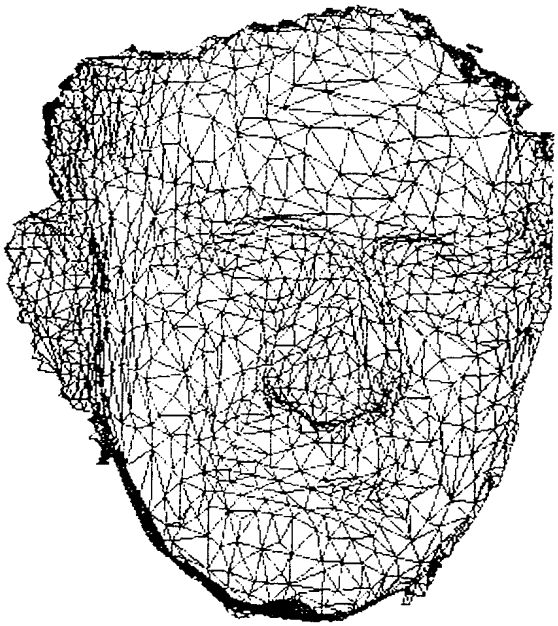


Fig. 5(a)

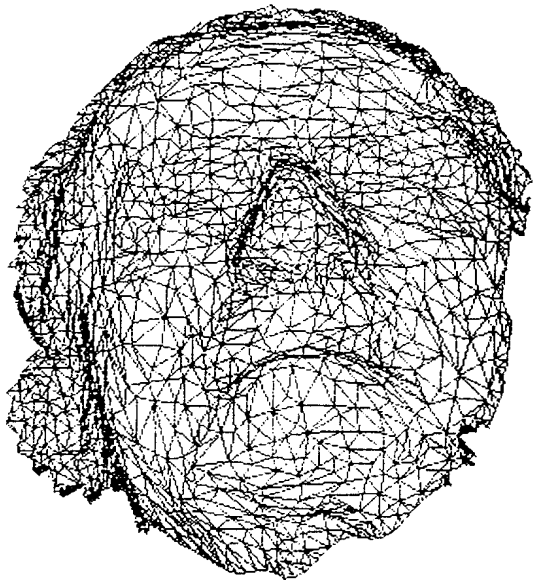


Fig. 5(b)

Fig. 6

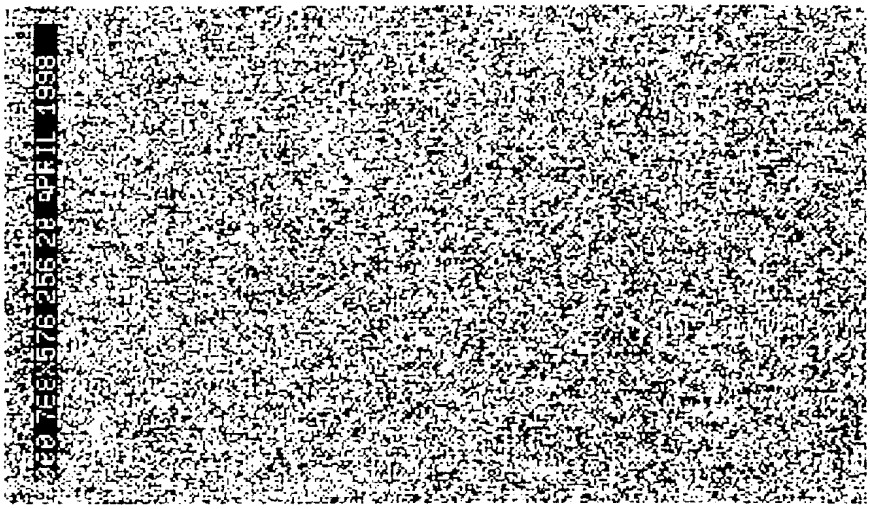


Fig. 7

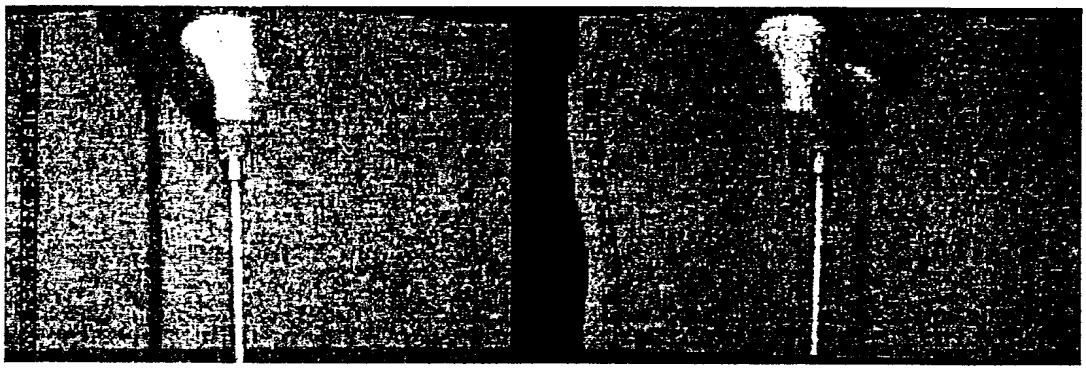


Fig. 8

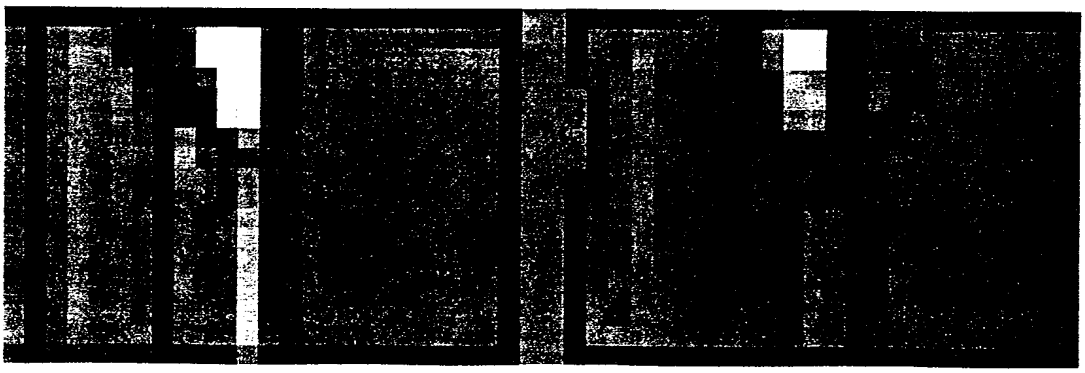


Fig. 9

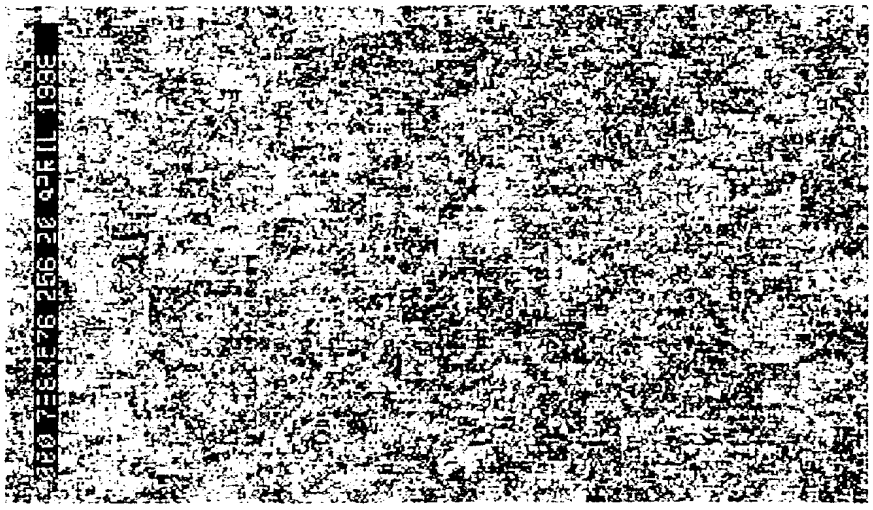


Fig.10

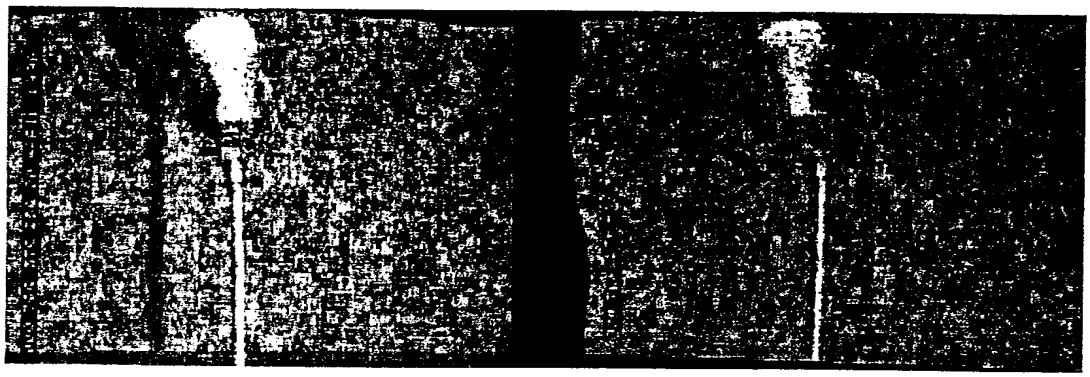


Fig.11

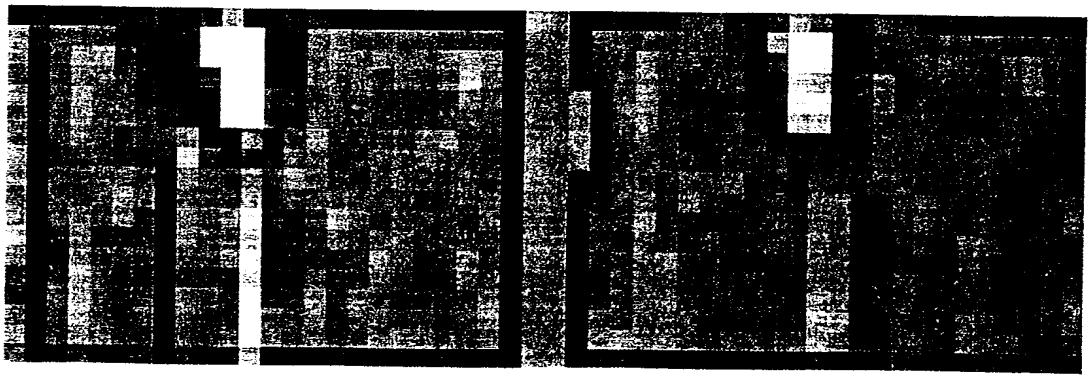


Fig.12

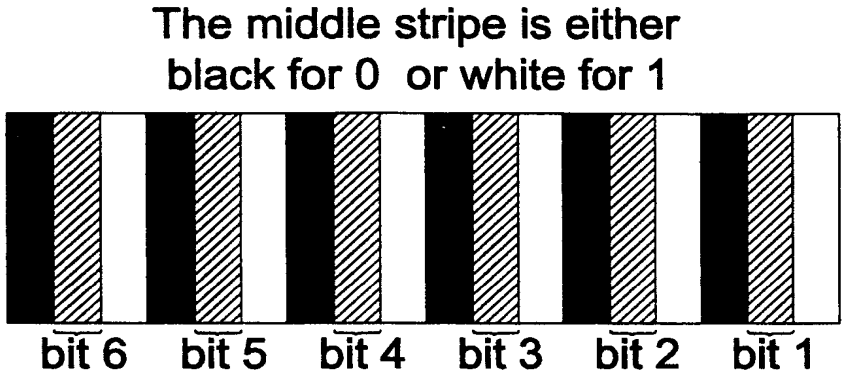

The middle stripe is either
black for 0 or white for 1
bit 6
bit 5
bit 4
bit 3
bit 2
bit 1

Fig.13

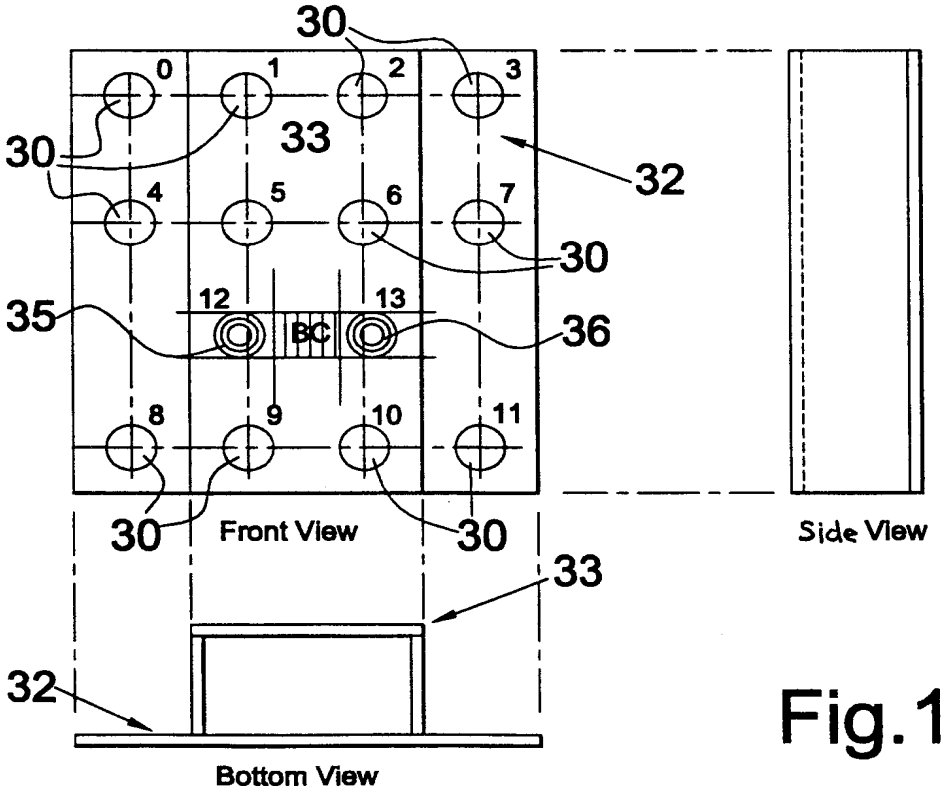

30
0
1
2
3
33
30
32
4
5
6
7
30
12
13
35
BC
36
8
9
10
11
30
Front View
30
Side View
33
32
Bottom View

Fig.14

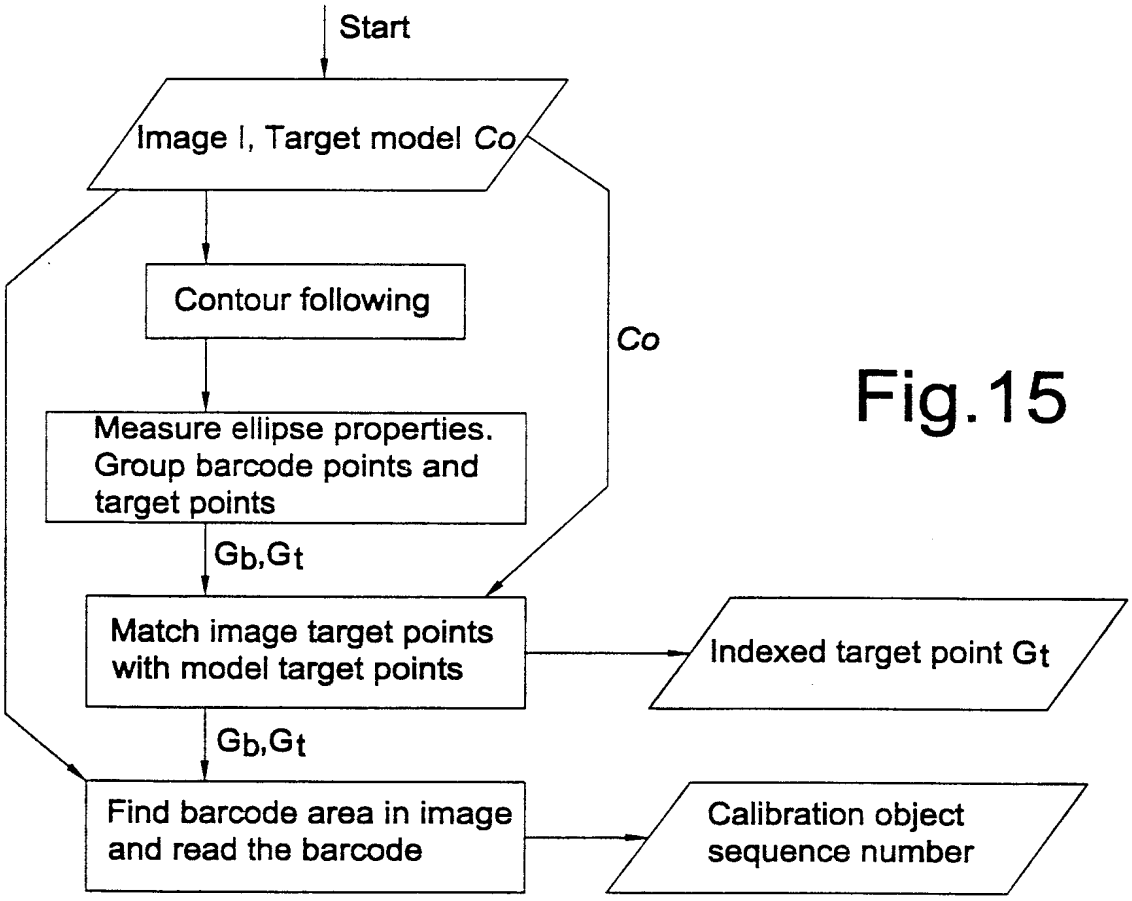

Start
Image I, Target model Co
Contour following
Co
Measure ellipse properties. Group barcode points and target points
Gb,Gt
Match image target points with model target points
Indexed target point Gt
Gb,Gt
Find barcode area in image and read the barcode
Calibration object sequence number

Fig.15

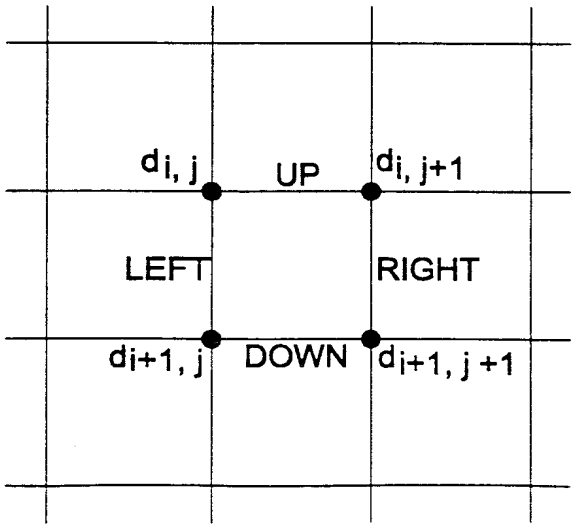
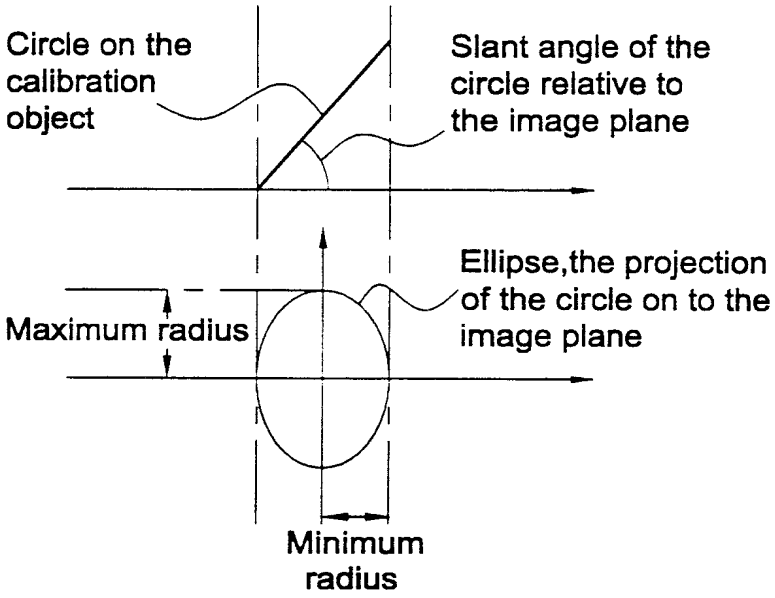
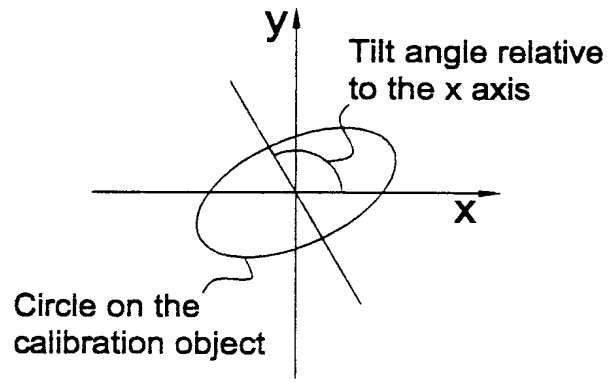

di, j
UP
di, j+1
LEFT
RIGHT
di+1, j
DOWN
di+1, j +1

Fig.16




Definition of the slant angle

Fig.17a




Definition of the tilt angle

Fig.17b

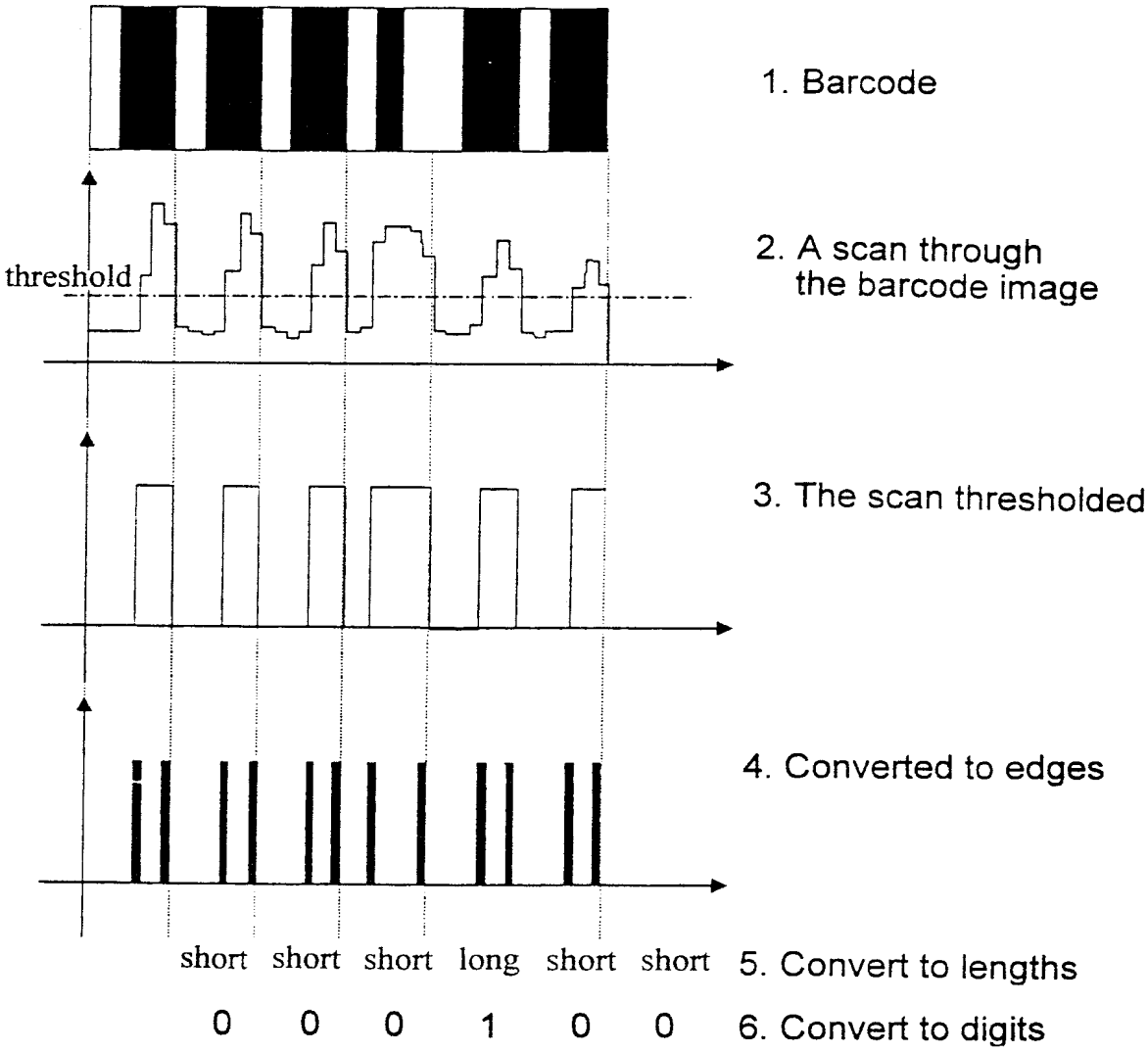

1. Barcode
threshold
2. A scan through the barcode image
3. The scan thresholded
4. Converted to edges
short short short long short short
5. Convert to lengths
0 0 0 1 0 0
6. Convert to digits

Fig.18

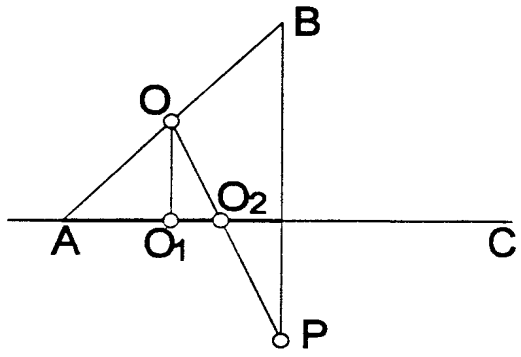

B
O
O2
A
O1
C
P

Fig. 19

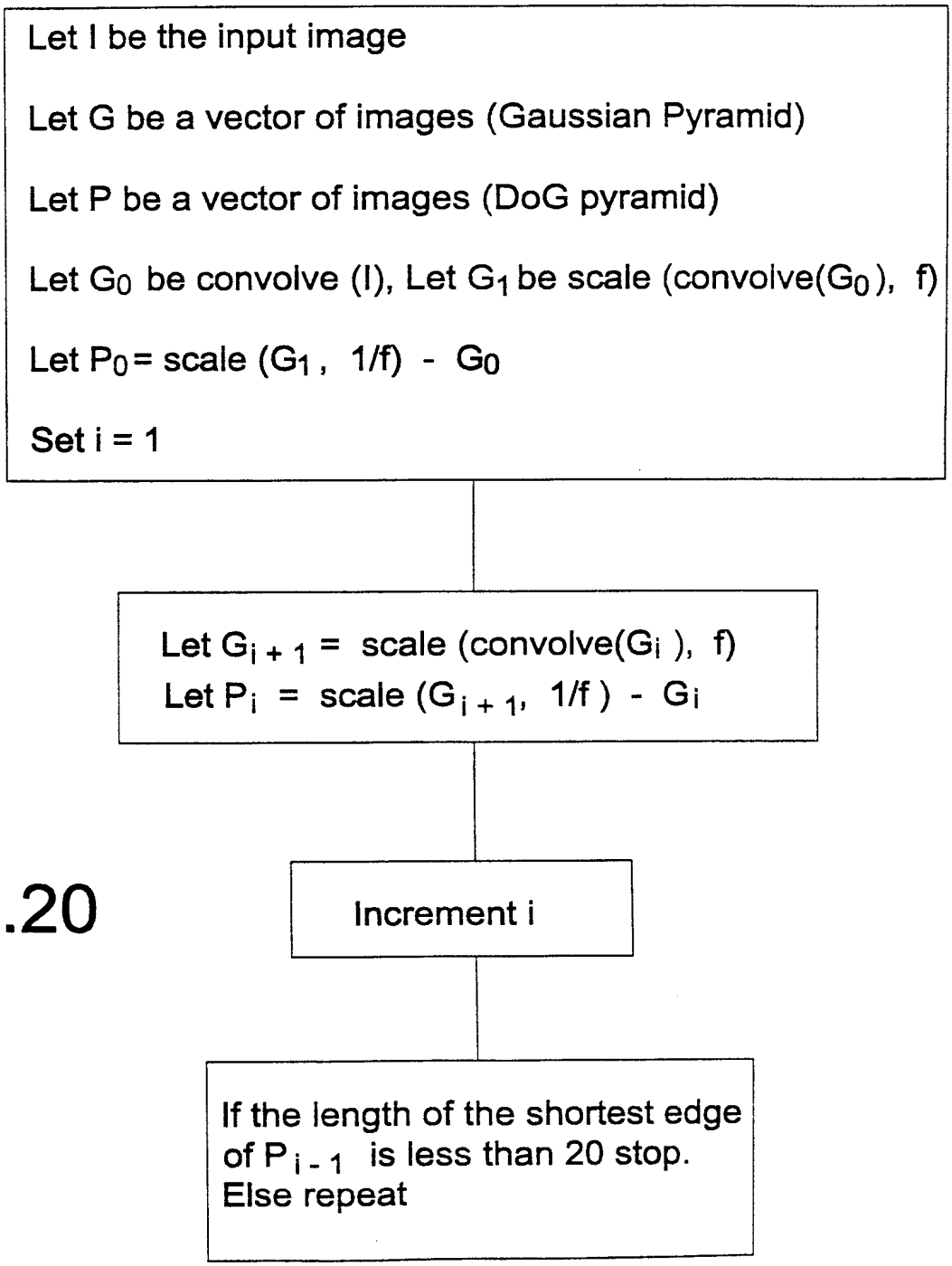

Let I be the input image
Let G be a vector of images (Gaussian Pyramid)
Let P be a vector of images (DoG pyramid)
Let G0 be convolve (I), Let G1 be scale (convolve(G0), f)
Let P0 = scale (G1, 1/f) - G0
Set i = 1
Let Gi + 1 = scale (convolve(Gi ), f)
Let Pi = scale (Gi + 1, 1/f ) - Gi
Increment i
If the length of the shortest edge of Pi - 1 is less than 20 stop.
Else repeat

Fig.20

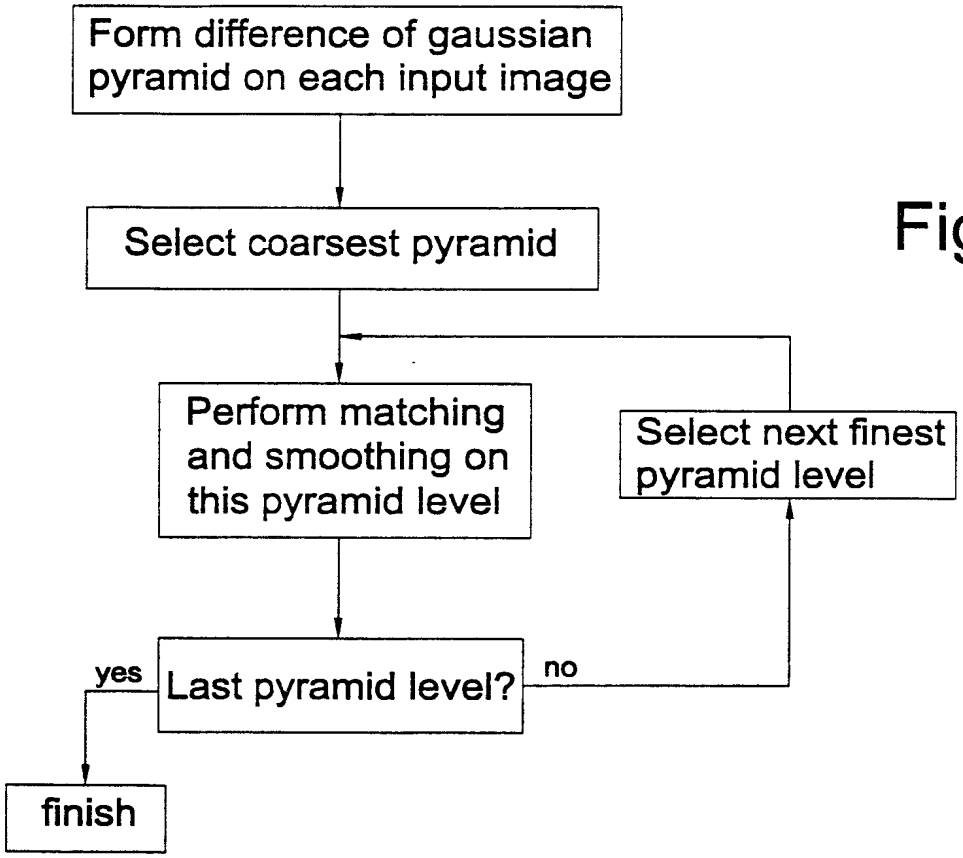

Form difference of gaussian pyramid on each input image
Select coarsest pyramid
Perform matching and smoothing on this pyramid level
Select next finest pyramid level
Last pyramid level?
yes
no
finish

Fig.21

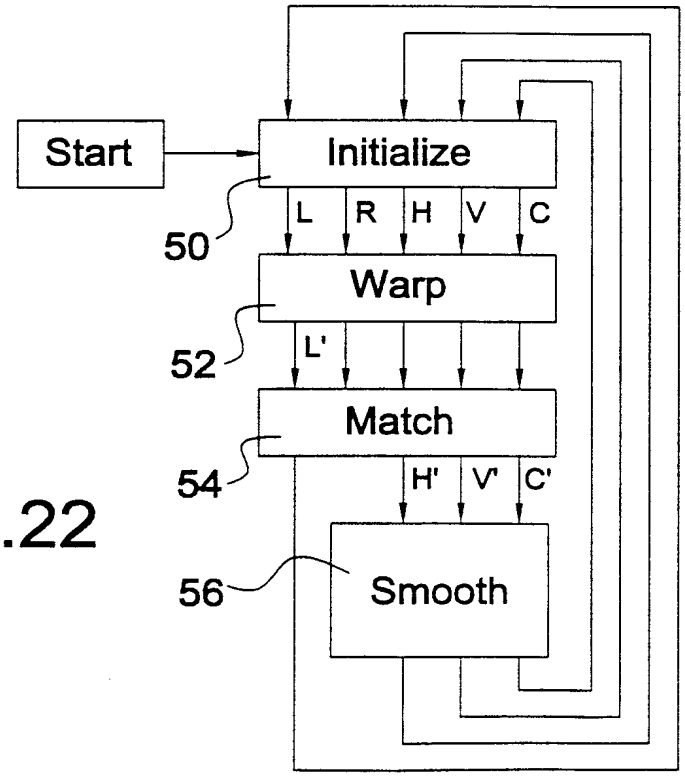

Start
Initialize
L
R
H
V
C
50
Warp
L'
52
Match
54
H'
V'
C'
56
Smooth

Fig.22

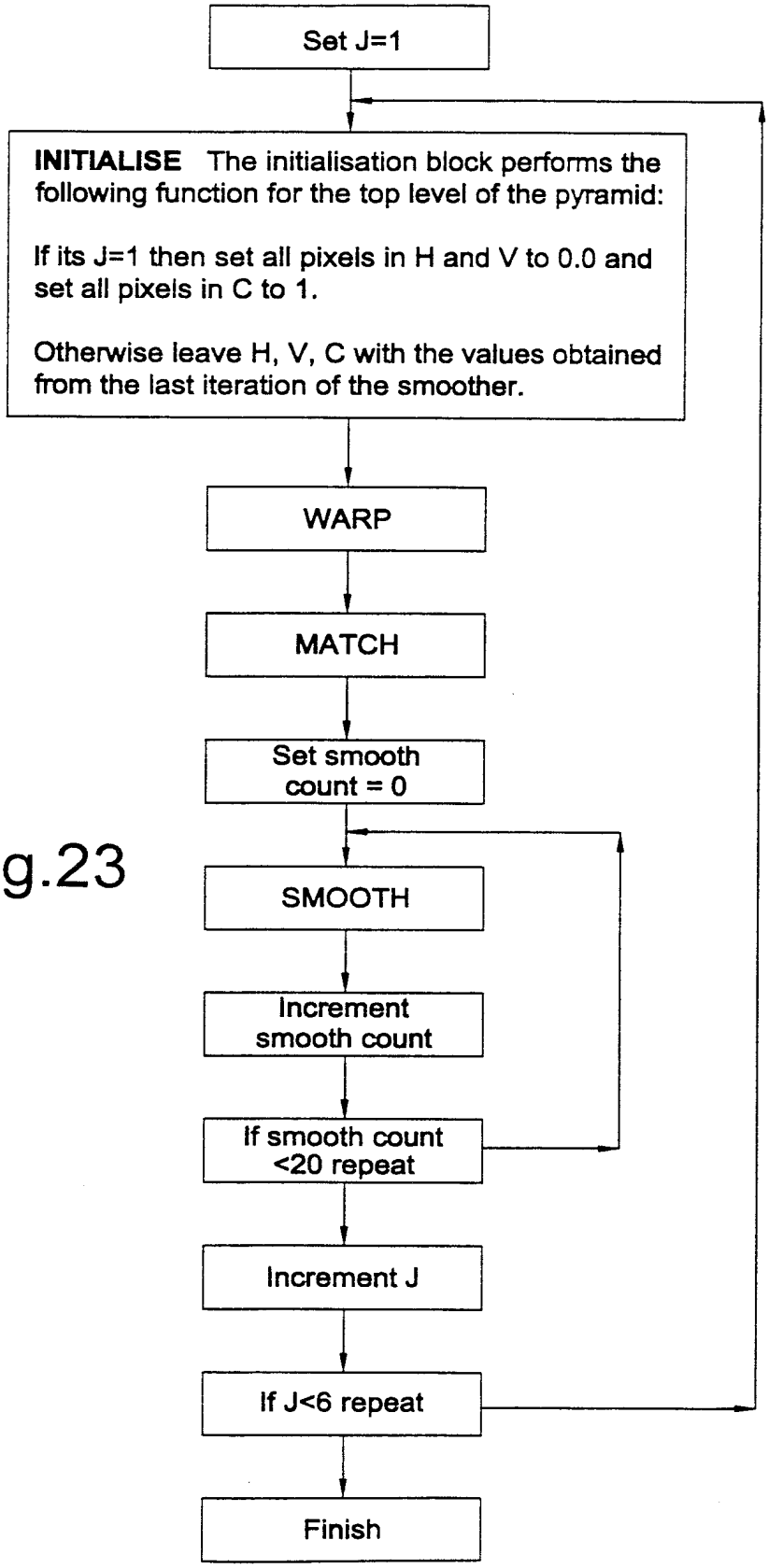

Set J=1
INITIALISE The initialisation block performs the following function for the top level of the pyramid:
If its J=1 then set all pixels in H and V to 0.0 and set all pixels in C to 1.
Otherwise leave H, V, C with the values obtained from the last iteration of the smoother.
WARP
MATCH
Set smooth count = 0
SMOOTH
Increment smooth count
If smooth count <20 repeat
Increment J
If J<6 repeat
Finish

Fig.23